Hardware Requirements

The hardware requirements for the NetRanger/Director are dictated by the Hardware requirements of HP OpenView. Consult the HP OpenView Installation documentation to ensure that your machine is powerful enough to run HP OpenView. In general, it is recommended that you use a dedicated machine that has at least 64 MB of RAM and at least 2 Gig of disk space.

Installing Solaris 2.4 or greater

Follow the directions in your Sun Solaris documentation to either install or upgrade to Solaris 2.4.

Installing HP OpenView 4.1 or greater

1. **Before attempting to install HP OpenView, ensure that the following parameters are set correctly:**

- IP Address
- Hostname
- Subnet mask
- Default gateway IP Address
- Default gateway Hostname
- system time and timezone

2. **Reboot the machine. Once the machine has rebooted, you should be able to ping your loopback address, ping your IP address, resolve your loopback address, resolve your IP address, and resolve your hostname. Also, the timezone should be correct. Do not go to the next step until TCP/IP is properly configured.**

CAUTION
HP OpenView will not install correctly if TCP/IP is improperly configured.

3. **Install HP OpenView 4.1 or greater on the Sun Solaris box (see the HP OpenView Installation Manual for details).**

4. **Add the following lines to the /.profile (please note the space between the "." and the "/":**

```
. /opt/OV/bin/ov.envvars.sh
export PATH=$PATH:$OV_BIN
```

Director Installation for Sun Solaris Systems

To install the NetRanger Director software on a Sun Solaris platform, follow these steps:

1. **Using "su", become the root user.**
2. **To load the OpenView environment variables, type the following command:**

```
. /opt/OV/bin/ov.envvars.sh
```

3. **If the OpenView user interface is running, stop it now by selecting Map..Exit from the OpenView menu. If other users have other copies of the user interface running and exported to other displays, ask them to shut down the user interface temporarily.**
4. **Put the NetRanger/Director tape in the tape drive if you have not already done so.**
5. **Go to the /tmp subdirectory by typing:**

```
cd /tmp
```

6. **The NetRanger/Director install tape should contain five compressed .tar files whose names have the following format:**

```
WGCnsx.<version>.<release>.<mod level>.<sys type>.tar.Z

WGCdrctr.<version>.<release>.<mod level>.<sys type>.tar.Z

WGCcfg.<version>.<release>.<mod level>.<sys type>.tar.Z

WGCsapd.<version>.<release>.<mod level>.<sys type>.tar.Z

JDK-<version>_<release>_<mod level>-<sys type>.tar.Z
```

The first file contains NSX-related files, and the other file contains the Director-related files.

7. **Untar these files using the following syntax (you must run this command for each of the five files):**

```
tar -xvf /dev/rmt/0m <filename>
```

Where <filename> is the name of the compressed tar file.

8. **Uncompress these files using the following syntax (you must run this command for each of the five files):**

   ```
   uncompress <filename>
   ```

   The files should now be uncompressed and should no longer have the ".Z" extension.

9. **Untar the uncompressed files, *except for the Java Development Kit*, using the following syntax:**

   ```
   tar -xvf <filename>
   ```

10. **Untar the Java Development Kit using the following commands:**

    ```
    mkdir -p /opt/SUNWjava
    cd /opt/SUNWjava
    tar -xvf /tmp/JDK-1_0_2-solaris2-sparc.tar
    mv java JDK
    cd /tmp
    ```

10. **Run the "package add" program by typing:**

    ```
    pkgadd -d .
    ```

11. **Select the "WGCnsx" product from the "available packages" list.**
12. **If this is a new installation, answer "yes" to overwrite an existing configuration files. If this is a re-installation, choose "no".**
13. **Answer "yes" to the question about "install suid programs".**
14. **Answer "yes" to run the script as root.**
15. **Once the WGCnsx installation process has completed, select the "WGCdrctr" product from the "available packages" list.**
16. **Answer "yes" to any questions the installation process might ask.**
17. **Select the "WGCcig" product from the "available packages" list.**
18. **Answer "yes" to any questions the installation process might ask.**
19. **Select the "WGCsapd" product from the "available packages" list.**
20. **Answer "yes" to any questions the installation process might ask.**
21. **After the installation procedure is complete, type "q" to quit.**
22. **The Director installation process creates an account for the user "netrangr". You must set a password for that user. To set the password, type:**

    ```
    passwd netrangr
    ```

19. **If */usr/nr/tmp* and */usr/nr/var* do not already exist, type the following to create them:**

```
mkdir /usr/nr/tmp

mkdir /usr/nr/var
```

20. **If this is a Solaris 2.4 installation, run the script below. If this is not a Solaris 2.4 installation, you do not need to run this script.**

```
/usr/nr/bin/postinstall.sh
```

21. **Examine the file */tmp/nrdirmap.install.out* to ensure that no errors occurred.**

The installation is now complete. Go to the section called "Post-Installation for HP-UX and Sun Solaris Systems."

Post-Installation for HP-UX *and* Sun Solaris Systems—Cleanup

1. **Start the HP OpenView daemons by typing the following:**

```
$OV_BIN/ovstart
```

   If the ovstart executable is not found, then the `$OV_BIN` environment variable is probably not set. To set the variable, please refer to the step on loading OpenView environment variables at the beginning of this section.

2. **To ensure that all OpenView daemons are running properly, type the following command:**

```
$OV_BIN/ovstatus
```

3. **Remove all NetRanger Director tar files from the /tmp directory using the "`rm`" command.**

4. **From /tmp, remove the WGC directories by typing the following:**

```
rm -rf WGC*.*
```

OPTIONAL

There is an OpenView daemon called *netmon* whose function is network mapping and availability monitoring. This daemon can be CPU-intensive. If you are only using the NetRanger/Director functionality, then it is not necessary to run this daemon.

To disable the *netmon* daemon, run the following commands:

```
$OV_BIN/ovstop netmon

$OV_BIN/ovdelobj $OV_LRF/netmon.lrf
```


5. **If necessary, set the DISPLAY variable in the appropriate .profile files.**

Post-Installation for HP-UX *and* Sun Solaris Systems—Background Process Configuration

This section describes configuration of the background processes that run on the Director machine. This is a very important set of steps, because if the background processes are not configured correctly, the Director processes will not know how to talk to the NSX machines in your network.

1. **Before editing the configuration files on the Director, it is a good idea to stop the OpenView user interface (using the Map..Exit menu), stop the OpenView daemons (by typing `$OV_BIN/ovstop`), and stop the NetRanger/Director daemons (by typing `/usr/nr/bin/nrstop`).**
2. **When you edit NetRanger configuration files, you should be logged on as the user** netrangr.

NOTE

All NSX boxes should have their */usr/nr/etc/destinations* file configured to send alarms to the *smid* process on the Director. This is very important. **The Director will not receive alarms if this is not done.**

(NOTE BOX) For specific information about changing the configuration files listed below, please refer to pages III-13-III-19

3. **All NSX boxes should list the proper name and address of the Director in the */usr/nr/etc/routes* file.**
4. **On the Director machine, modify the */usr/nr/etc/auths* file to list the Director machine itself.**
5. **On the Director machine, ensure that the */usr/nr/etc/hosts* file matches the */usr/nr/etc/hosts* files on the NSX machines. The only difference should be that the *localhost* entry in the Director's file should match the Director's *hostId/orgId* pair.**
6. **On the Director machine, ensure that the */usr/nr/etc/destinations* file lists all of the NSX boxes that are sending data to the Director machine.**
7. **On the Director machine, ensure that the */usr/nr/etc/routes* file matches the routes files on the NSX boxes.**
8. **On the Director machine, ensure that the */usr/nr/etc/daemons* file has a listing for *nr.smid*. If there is an entry for *nr.sensord* in this file, ensure that it is commented out. (Use the # key to comment out a line.)**
9. **On the Director machine, ensure that the */usr/nr/etc/smid.conf* file lists the *loggerd* daemon as a *dupDestination*.**

The installation should now be complete.

To start the NetRanger Director, follow these steps:

1. **Log on as root and start the OpenView daemons using the `ovstart` command.**
2. **Log on as netrangr and start the NetRanger daemons using the `nrstart` command.**
3. **Start the user interface by typing the following:**

   ```
   $OV_BIN/ovw &
   ```

Post-Installation for HP-UX *and* Sun Solaris Systems—Configuring the User Interface

The following steps only need to be done once.

1. **Start the OpenView user interface if you have not already done so by typing**

   ```
   ovw &
   ```

2. **Double-click the NetRanger icon.**
3. **Select the menu option Map..Maps..Describe/Modify.**
4. **Press the Propagate Most Critical button under the Compound Status Heading.**
5. **Choose OK.**
6. **Select the menu option Map..Submaps..Set This Submap As Home.**
7. **Select the menu option Map..Submaps..Describe/Modify.**
8. **Under the Background Graphics: heading, press the Browse... button.**
9. **From the pop-up list, select the background graphic of your choice. (The usastates.gif is a popular choice). You could also create a custom GIF file with any graphics program, and use that GIF file as an OpenView submap background.**
10. **Choose OK, and then choose OK again.**

The following steps should be done every time a new NetRanger/NSX box is added to your system:

1. **Add the new NSX information to the */usr/nr/etc/*destinations file.**
2. **Add the new NSX information to the */usr/nr/etc/*hosts file.**
3. **Add the new NSX information to the */usr/nr/etc/*routes file.**
4. **Add the NSX icon to the OpenView map using the following steps:**
   - Select the **Edit..Add Object** menu function.
   - Click on the **Net Device** icon. Several icons will appear in the bottom of the window.
   - Drag the **NSX 2000** icon to the NetRanger submap (the submap containing the Director icon) by pressing and holding the **middle** mouse button while positioning the mouse pointer over the **NSX 2000** icon. An **Add Object** window should appear.

5. **Select** NetRanger/Director **from the list, and press the** Set Object Attributes **button.**
6. **In the** hostname **field, enter the name of the NSX box exactly as you entered it in the *letc/nr/etc/hosts* file.**
7. **Press the** Verify **button. If you entered the hostname correctly, the Organization and Host Ids should have been filled in for you.**
8. **Once the** hostname, Organization ID, **and** Host ID **are correct, choose** OK.
9. **Press the "Set Selection Name" button. Choose the selection name from the list and then press OK in the Selection Name window. Press the OK button in the Add Object Window.**
10. **You should see the** NSX **icon turn green. If you double-click on the NSX icon, you should see icons that represent the processes running on that machine.**

NOTE

There are two files, */usr/nr/bin/nr.envvars.sh* and */usr/nr/bin/nr.envvars.csh*, that contain environment variables needed for running the NetRanger Director. All users who want to run the NetRanger Director should source these files in either their .profile or .login files (depending on which shell they use).

For example, the user **netrangr** who uses Korn Shell should add the following line to the .profile in */usr/nr* to load the proper environment variables when netrangr logs in:

```
. /usr/nr/bin/nv.envvars.sh
```

Data Privacy Facility

Included as part of the NetRanger package, the NSC BorderGuard, comes with powerful encryption capability that can establish a Virtual Private Network (VPN) between geographically remote sites over public networks.

At a minimum, WheelGroup recommends that you use encrypted sleeves to encrypt all traffic between remote networks using the NetRanger NSX Sensor and the primary network that should be running both the NetRanger NSX Sensor and the Director. You do not necessarily have to use encrypted links between any internal NSXs you may be using for which traffic is only crossing your internal private networks.

A good method to determine a necessity for encrypted sleeves is to trace the path of alarms and traffic traveling between two NSX systems and consider the security of the networks that cross. If you feel comfortable with those networks, it is probably not necessary to use encryption and incur the performance reduction on traffic traveling within the sleeve.

IV Operating the Director

Overview

The NetRanger Director is the Graphical User Interface (GUI) to NetRanger. The NetRanger Director (also called "the Director") has four main functions:

- provides in real time a graphical, intuitive display of information pertaining to network security violations;
- displays a hierarchical map of the remote NetRanger software and hardware (the Sensor processes and the NSX hardware, for example) that send security notifications to the Director;
- provides utilities for configuration of the remote NetRanger applications; and
- provides utilities to query the database of historical security events.

The Director uses popular network management platforms like HP OpenView and IBM NetView for AIX to display network security information. As a result of this integration, network management personnel do not have to learn multiple-user interface applications and paradigms to perform different network management tasks.

When a process on a remote NSX machine detects a security violation, a notification (called an "event") is sent from the NSX machine to the Director machine. The Director ensures that the machine and application that generated the event are represented on the graphical map, and then, if the event's severity level exceeds a user-definable threshold, the Director creates an Alarm icon on the map. The color of the Alarm icon is based on the severity of the event. The Application and Machine icons also change color, so it is easy to determine at a glance which machine detected the problem. With a few mouse clicks details about the Alarm (source and destination IP address, for example) can be displayed. Location functions can be used to locate Alarms with specific properties.

Once an Alarm is diagnosed and addressed, the user can delete the Alarm icon from the User Interface. The Application and Machine icons go back to their previous state.

Architecture

The NetRanger Director is not a single computer program, but is rather a *set* of applications and background processes that work with a network management platform. The diagram below illustrates the data flow between the processes in NetRanger.

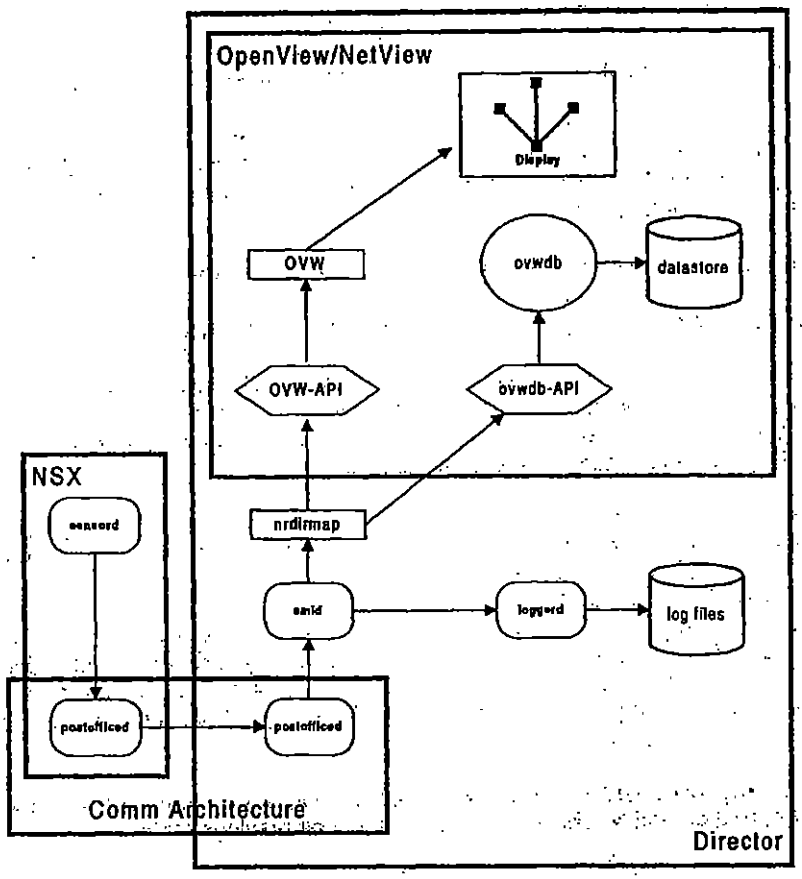


Figure IV-1: The NetRanger Director Architecture

In the diagram above, ovals represent background processes, squares represent foreground applications, cylinders represent datastores, hexagons represent APIs, and lines represent the flow of event data. Note that *ovw* and *ovwdb* are part of OpenView/NetView, *nrdirmap*, *smid*, and *loggerd* are part of the Director, and *sensord* is part of the NSX. Also note that the NSX and the Director both contain *postofficed* processes.

When the *sensord* process detects activity of interest, it generates an event that is sent via the *postofficed* daemons to the *smid* daemon on the Director machine. The *smid* daemon passes the event information to the *loggerd* daemon, which logs the information, and to *nrdirmap*.

nrdirmap looks at the severity level of the event. If the event severity exceeds a user-specified level, then *nrdirmap* tells *ovw* to draw an alarm icon. *nrdirmap* also tells *ovwdb* to create an alarm database object in the OpenView/NetView datastore.

Basic Director Functions

Starting the Director

NetRanger/Director is composed of three separate subsystems:

- The NetRanger background processes
- The network management platform background processes
- The network management platform user interface

NOTE

These subsystems should be started in the order listed above in order to ensure proper operation of the Director.

Starting the NetRanger Background Processes

The NetRanger background processes are configured to start automatically when the machine is rebooted, so in most cases, you will not need to manually start these processes.

In the event that you must start them manually, follow these steps:

1. **Log in as someone in the group `netrangr`, and then type:**

   `nrstart`

2. **If the executable is not found, then either type the fully qualified name**

   `(/usr/nr/bin/nrstart)`

   or put `/usr/nr/bin` in your path.

Starting the Network Management Background Processes

Like the NetRanger background processes, the network management platform background processes are configured to start automatically when the machine is rebooted, so in most cases, you will not need to manually start these processes.

In the event that you must start them manually, follow these steps:

1. **log in as `root` and then type:**

   `ovstart`

   If the executable is not found, then the subdirectory that includes the network management binaries is probably not in your path (the location for OpenView binaries is `$OV_BIN`, and the location for NetView binaries is `/usr/OV/bin`). Consult your network management documentation if you have difficulty starting the network management background processes.

Starting the Network Management User Interface

To start the Director's network management user interface, follow these steps:

1. **If you use HP OpenView, log in as a user that belongs to the group `netrangr` and then type:**

   ```
   ovw &
   ```

   If you use IBM NetView for AIX, log in as a user that belongs to the group `netrangr` and then type:

   ```
   nv6000
   ```

NOTE
The *nrdirmap* program will start automatically when you bring up the network management user interface. You will never have to manually start *nrdirmap*.

Stopping the Director

To stop the Director, stop the subsystems in the *opposite* order in which they were started.

Stopping the Network Management User Interface

1. **If you use HP OpenView, select the menu option**

   ```
   Map..Exit
   ```

2. **If you use IBM NetView for AIX, select the menu option**

   ```
   File..Exit
   ```

Usually, you will only want to close the user interface. In most circumstances, you will not want to close the background processes. If you do want to close the background process, follow the steps below.

1. **Log in as `root` and then type**

   ```
   ovstop
   ```

   If the executable is not found, then the subdirectory that includes the network management binaries is probably not in your path (the location for OpenView binaries is `$OV_BIN`, and the location for NetView binaries is `/usr/OV/bin`). Consult your network management documentation if you have difficulty starting the network management background processes.

Stopping the NetRanger Background Processes

To stop the NetRanger background processes, follow these steps:

1. **Log in as someone in the group `netrangr` and then type:**

   ```
   nrstop
   ```


2. **If the executable is not found, then either type the fully qualified name**

   `(/usr/nr/bin/nrstop)`

   or put `/usr/nr/bin` **in your path.**

Checking the Status of the Director Processes

To check the status of all Director processes, follow these steps:

1. **To ensure that the network management background processes are running correctly, type**

   `ovstatus`

2. **To ensure that the NetRanger background processes are running correctly, type**

   `nrstatus`

   If either of these executables cannot be found, check your path.

Understanding the Director's Submap Hierarchy:

When you double-click on a symbol, a submap is opened. This submap could have many symbols on it, and these symbols could be double-clicked to reveal more submaps, each with many symbols. This set of descending submaps can be thought of as an upside-down tree with more and more branches. This upside-down tree structure of submaps and symbols is called the "submap hierarchy".

Traversing the submap hierarchy that nr.dirmap creates is easy once you understand the following structure:

This type of submap...	can contain these symbols:
Root	Collection
Collection	Machines (NSX and Director)
	Collections
	Connections
Machine	Applications
Application	Alarms

NOTE

Note that Connections and Alarms do not have submaps. They represent the "leaves" in the submap tree.

Figure IV-2 illustrates the Root submap. It is the highest level submap in the hierarchy. The root submap has no "parent submap". On the root submap, there should be a symbol representing a Collection of machines.

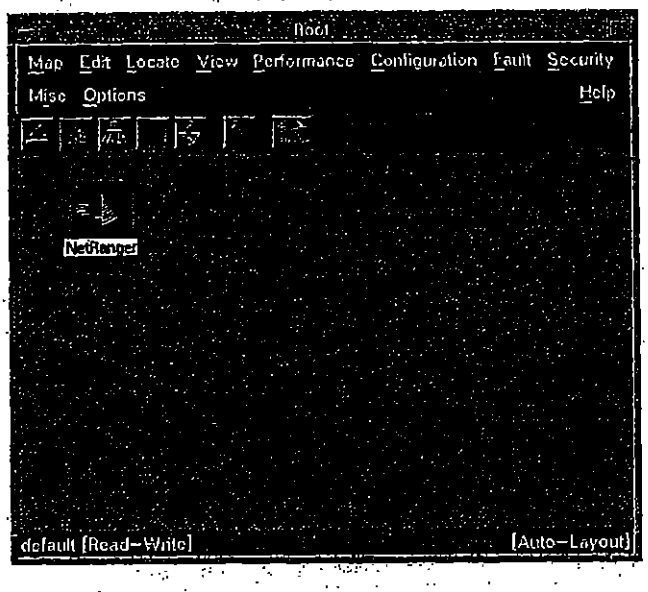


Figure IV-2: The NetRanger Director Submap

When you double-click on a Collection symbol like the one shown in Figure I-3, a Collection submap is displayed. A Collection submap can have NSX Machines, the Director Machine, other Collection symbols, and Connections between Machines. The Collection submap shown on the following page only contains a Director machine symbol.

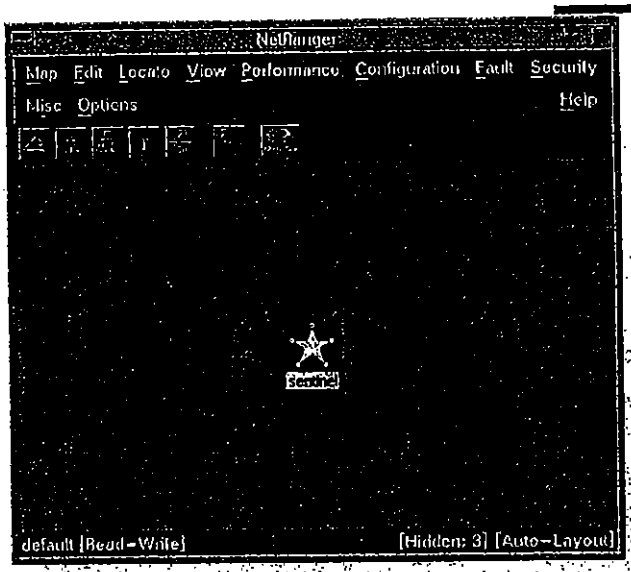


Figure IV-3: A Collection Submap

When you double-click on a Machine symbol, a Machine submap is displayed. A Machine submap contains symbols that represent the different applications running on the machine.

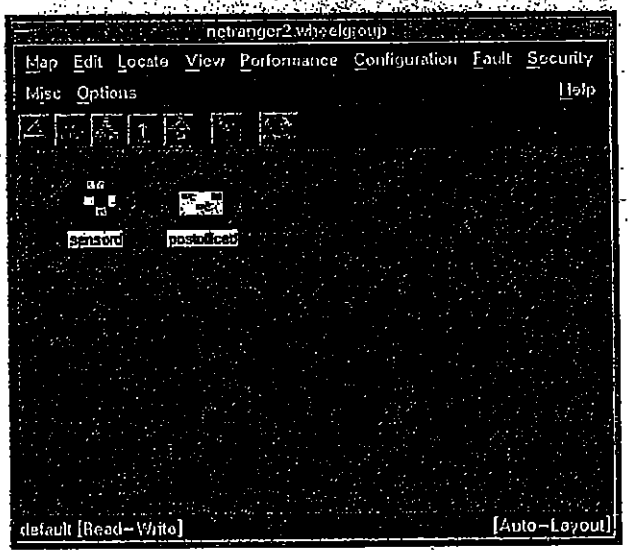


Figure IV-4: A Machine Submap

When you double-click on an Application symbol, an Application submap is displayed. An application submap contains alarms that that application had generated. For instance, if the *sensord* application for a machine generates an event and sends it to the Director, the Director will draw an alarm icon on the submap belonging to that machine's application.

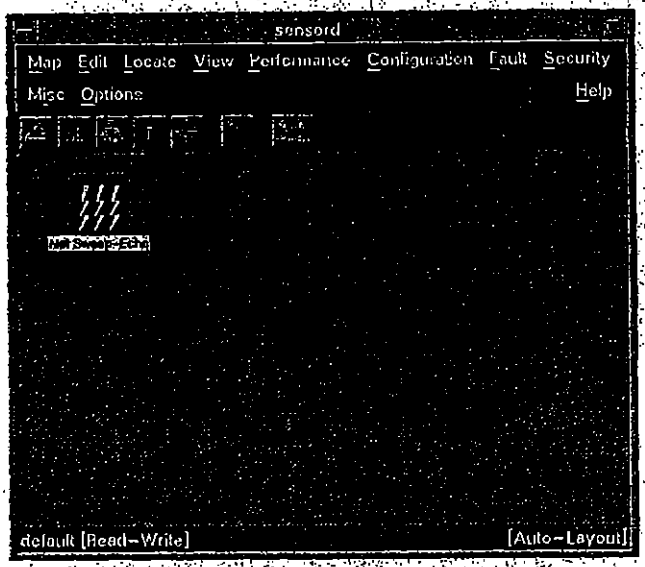


Figure IV-5: An Application Submap

In the above Application submap, an Alarm is displayed that indicates that a "Net Sweep" has occurred. If an Application has generated no Alarms, then a special Alarm called an "OkAlarm" will be displayed that indicates that the Application has no unresolved Alarms. The "OkAlarm" is shown on the following page.

Figure IV-6: An OkAlarm

Adding Entities

In general, there are four types of icon symbols of interest: alarms, applications, machines, and collections.

Alarm symbols, at the bottom of the submap hierarchy, can only be created by the *nrdirmap* application. An alarm symbol is created whenever an event that exceeds a user-defined threshold is received. There is no way for a user to manually create an alarm symbol.

There are two ways that Application and Machine symbols can be created. First, if an application or host from which an event emanates is not already represented in the map, then *nrdirmap* will create the symbols for you.

If you do not want to wait until an alarm comes in to have a machine or an application represented in a map, you can add the symbols manually. The next two sections describe how to add machines and applications.

Manually Adding an NSX Machine Symbol

To manually add an NSX machine symbol, follow these steps:

1. **Double-click on a Collection symbol to open the Collection submap (the symbol on the root submap labeled "NetRanger" is a Collection symbol). Machines can only be added to Collection submaps! Do not try to add a Machine to a non-Collection submap.**
2. **Select the Edit..Add Object menu function. The Add Object Palette will appear.**
3. **Click on the Net Device icon. Several icons will appear in the bottom of the palette (see picture below).**

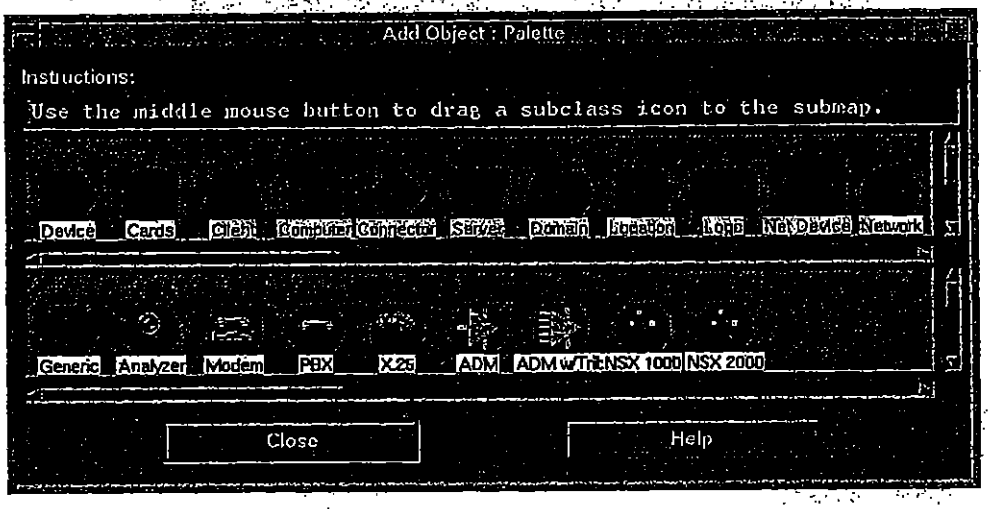


Figure IV-7: The Add Object Palette

4. **Position the mouse pointer over the NSX 2000 icon, press and hold the middle mouse button, and drag the NSX 2000 icon to the NetRanger submap (the submap that has the Director icon on it). An Add Object window should appear.**
5. **Select NetRanger/Director from the list, and press the Set Object Attributes button.**

6. In the hostname field, enter the name of the NSX box exactly as you entered it in the /usr/nr/etc/hosts file.
7. Press the Verify button. If you entered the hostname correctly, NetRanger/Director will populate the Organization and Host Id fields for you.
8. Once the hostname, Organization ID, and Host ID are correct, press OK.
9. In the Selection Name field, enter a few random characters, and then press OK. (In a future release, NetRanger/Director will populate this field for you, and you will not need to enter any text here.)

Manually Adding an Application Symbol

1. Double-click on the NSX Machine to which you wish to add the application. Applications can only be added to Machine submaps! Do not try to add an Application to a non-machine submap!
2. If the Add Object Palette is not already displayed, bring it up by selecting the Edit..Add Object menu function.
3. From the Add Object Palette, click on the WGC Application icon. Several icons should appear in the bottom of the window.
4. Using the same technique described above, drag the application icon to the NSX submap.
5. Select NetRanger/Director from the list, and press the Set Object Attributes button.
6. In the Application Name field, enter the name of the application you just moved to the submap. (In a future release, NetRanger/Director will populate this field for you, and you will not need to enter any text here.)
7. Press the Verify button. If you entered the application name correctly, the Application ID should have been filled in for you.
8. Press OK.
9. In the Selection Name field, enter a few random characters, and then press OK. (In a future release, NetRanger/Director will populate this field for you, and you will not need to enter any text here.) You should see the Application icon turn green, because a green OkAlarm will be created automatically in the submap of the Application you just added.

Manually Adding an NSX Collection

The **Top-Level NSX Collection** (the entity that appears on the Root submap labeled **NetRanger**) is created for you. This is the only Collection that can appear on the root submap. Do *not* try to create additional Collections on the root submap.

NSX Collections are used to customize, or "partition," the map. NSX Collections are good tools to use to group machines into logical units. See the section of this guide called *How to Customize a Map* for more information about specific uses of NSX Collections.

Follow these steps to add a NSX Collection:

1. **Double-click on the NSX Collection's submap that you want to add the new NSX Collection.** NSX Collections can only be added to NSX Collection submaps! **Do not try to add an NSX Collection to a non-NSX Collection submap!**
2. **If the Add Object Palette is not already displayed, bring it up by selecting the** Edit..Add Object **menu function.**
3. **From the** Add Object:Palette, **click on the** Location **icon. Several icons will appear in the bottom of the window.**
4. **Using the same technique described above, drag the symbol that has the WheelGroup Logo to the NSX submap.**
5. **Select** NetRanger/Director **from the list, and press the** Set Object Attributes **button.**
6. **In the** NSX Collection Name **field, enter the name of the NSX Collection you just moved to the submap. This name can be any unique string. For example, "New York," "Building 162," or "10.1.1 Machines" would be legitimate NSX Collection Names.**
7. **Press the** Verify **button.**
8. **Press** OK.
9. **In the** Selection Name **field, enter a few random characters, and then press** OK. **(In a future release, NetRanger/Director will populate this field for you, and you will not need to enter any text here.)**

Modifying and Viewing Entity Attributes

Once an entity has been created (either by a user or by *nrdirmap*), it is often helpful to view a listing of the entity's attributes. Some attributes can be edited by the user (for instance, a machine's Point of Contact), and other attributes are read/only (for instance, an alarm's Date). OpenView/NetView provide visual indication of which fields are changeable and which are not.

To display an entity's attributes, select the entity with the mouse pointer, and then select the menu function **Edit..Modify/Describe..Object**. You can also select the entity and type *ctrl-o*. A pop-up window will appear. Select **NetRanger/Director** from the list of applications, and then press the **Configure** button.

Different entities have different attributes, so each entity will be discussed separately.

NSX Collection Attributes

The NSX Collection Name is the single attribute of a NSX Collection. If you change the Name, and then press "OK" on the appropriate screens, the NSX Collection's symbol label and submap name will change to reflect the new name.

Machine Attributes

Machines have four attributes: **Organization ID, Host ID, Hostname,** and **Point Of Contact.** The hostname and point of contact are editable, but the **Org ID** and **Host ID** are not. If you need to change an Org or Host ID, the best thing to do is to delete the machine and then re-add the machine with the correct IDs. If the hostname is changed, the Machine's symbol label will be the part of the hostname up to the first dot ("."), and the submap name will be the entire hostname.

If you want to store more information about the Point of Contact than the single field will contain, there are two things you can do. First, you can use the **Object Comments** field to store the additional point of contact information. Second, you could put the point of contact information in a separate trouble ticketing system.

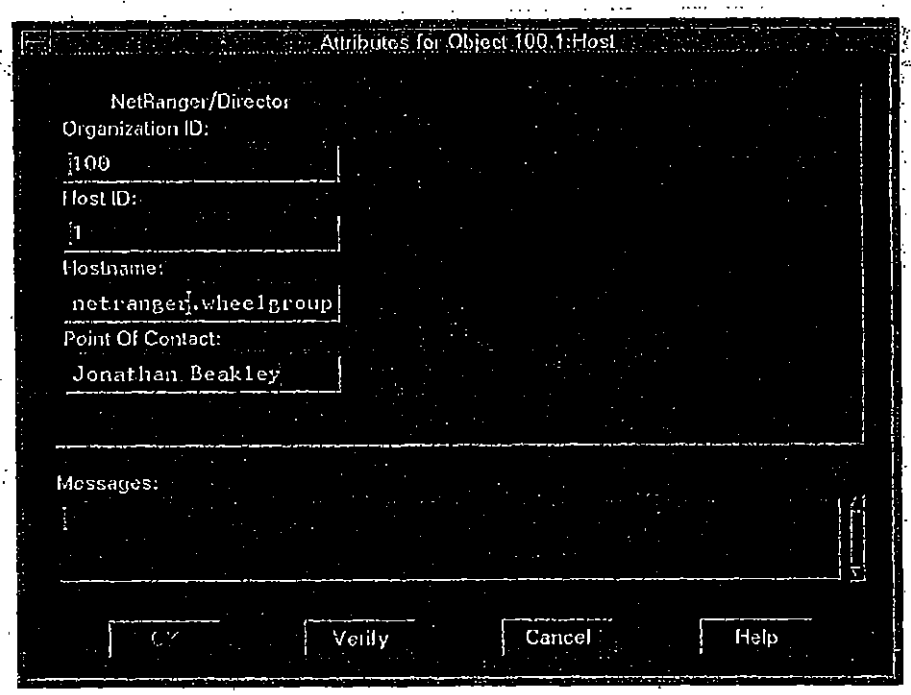


Figure IV-8: Object Attributes Window

Application Attributes

Machines have six attributes: **Application Name, Minimum Marginal Status Severity, Minimum Critical Status Severity, Organization ID, Host ID,** and **Application ID.** The application name and status severity fields are editable, but the ID fields are not. If you need to change an ID, delete the application and then re-add it with the correct IDs. If the application name is changed, the Application's symbol label will change to match the new application name, and the Application's submap name will reflect the new name, too (the format for the Application submap name is **<hostname>:<application name>**).

The **Minimum Marginal Status Severity** describes the lowest severity status an event can have before a marginal (yellow) alarm is created to represent that event. For example, if the minimum marginal status severity is 3, and a severity 2 alarm comes in, then no alarm entity will be created.

NOTE

The higher the severity level, the more severe the alarm. Currently, severity 6 is the highest severity level assigned by the *sensord* daemon.

The **Minimum Critical Status Severity** describes the lowest severity an event can have before a critical (red) alarm is created to represent that event. For example, if the minimum critical status severity is 3, and a severity 4 alarm comes in, then a critical alarm will be created.

NOTE

If you change a status severity value, only events generated **after the change** will be affected. If you increase a threshold severity level from 2 to 3, *nrdirmap* will not remove any existing level 2 alarms from the application's submap. Also, if you decrease a threshold severity level from 3 to 2, nrdirmap will not check historical log files and create alarm icons for severity level 2s that may have occurred in the past. Note that Connections and Alarms do not have submaps.

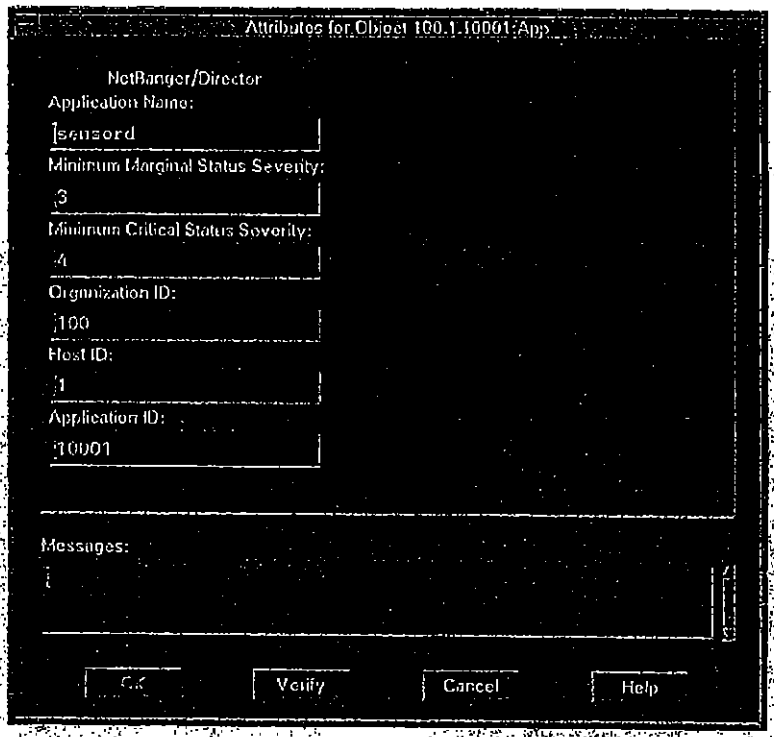


Figure IV-9: sensord Attribute Information

Alarm Attributes

Alarms have many attributes: **Name, Severity, Source Port, Destination Port, Source Address, Destination Address, Router Address, Date, Is Source Address Protected, Is Destination Address Protected, Details, Signature ID, Subsignature ID, Organization ID, Host ID, Application ID, Instance ID.** All alarm attributes are read-only.

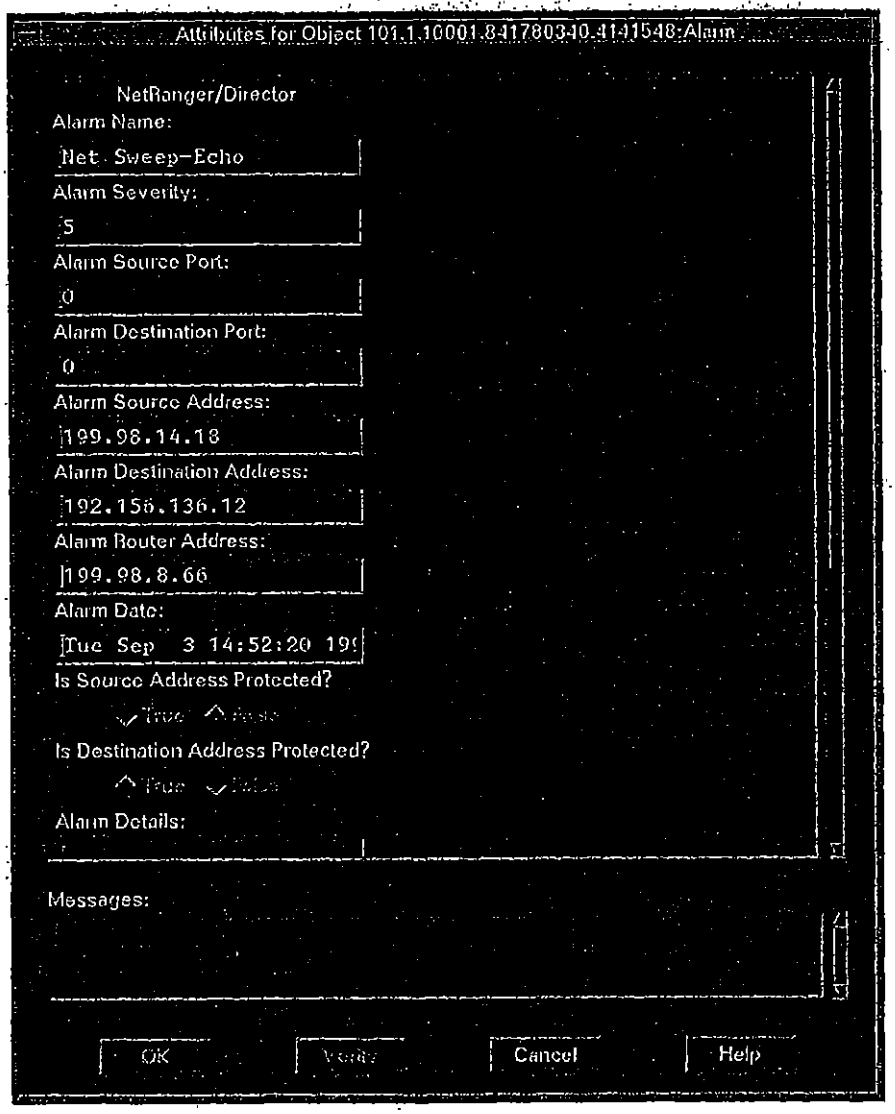


Figure IV-10: Alarm Event Attributes

The special **OkAlarm** that indicates that an application has no unresolved alarms has only four attributes: **Date, Organization ID, Host ID,** and **Application ID.** All these fields are read/only. The Date field specifies the time at which the **OkAlarm** was created. This gives a lower boundary to the last time that the application in question detected an attack.

Deleting Entities

When you want to remove a symbol (and its corresponding database object), you must select the symbol and then choose the **Edit..Delete Object..From *All*** Submaps menu option. The most common usage of the delete function is deleting an alarm symbol once the potential hacking attempt has been diagnosed and resolved.

There are rules governing the deletion of symbols that help prevent the accidental removal of alarms and other symbols. One general rule to remember is this:

NOTE
Applications and Machines can NOT be deleted until ALL of their alarms have been deleted.

This forces the user to go into the submap containing the alarms and specify that it is OK to delete the alarms. This helps prevent a hacking attempt from going unnoticed.

Once an application or host has had all of its alarms resolved (and deleted) you are free to delete the application or machine.

NOTE
If you delete an application or machine, and then an event is received for that machine, the machine will be redrawn on the map. In a case like this, it might be better to *hide* the machine (see the description of the Hide function, below).

Because it would be very easy to accidentally delete large groups of machines, non-empty Collections cannot be deleted. If you have a Collection that contains many machines, and you want to delete the Collection, you must go into the Collection submap and delete all of the machines first (and of course, the machines must have their alarms deleted before the machines can be deleted). Once you have emptied the collection submap, you can then delete the Collection.

NOTE
Never use the Delete Submap function! *nrdlmap* does *not* support this function. Always use the **Delete Object** function to delete entities!

How to Customize a Cap

NSX Collection entities can be used to customize, or "partition" a map. If the number of NSX machines you are monitoring is too great to represent on a single submap (for instance, the Top-Level NSX Collection submap), you can create additional Collections, and then add Machine icons to those Collection submaps. This allows you to create a hierarchical grouping of machines.

For example, if you had 25 NSX machines in Los Angeles, and 35 machines in New York, you could create an "LA Collection" entity and a "NY NSX" Collection entity. You could then add the NY NSX Machines to the NY Collection, and then add the LA NSX Machine to the LA Collection. This allows you to have fewer symbols per submap, which makes locating symbols and diagnosing problems faster and easier.

NOTE

To put a machine in a collection, you must use the **Add Object** function. If a machine is already represented on the map, and you want to move the representation (the symbol) from one collection to another, you must delete the machine and then re-add it. ***nrdlrmap* does *not* support the "Cut and Paste" functionality!** Use of the Cut and Paste functionality on *nrdlrmap* entities will yield unexpected results. *You must delete a machine and then re-add it to move the machine symbol from one Collection to another.*

Viewing Event Lists

To view an ASCII list of the latest events that have been generated for a given application or machine, simply select either an Application or a Machine symbol from the map, and then choose the menu option **Security..Show Current Events.**

This will execute a program that parses the log files in /usr/nr/var, looking for *all* events for the entity selected. Please note that this will include events that may be below the threshold for creating alarms.

Also note that this window is dynamically updated as new events come in. This is why the "hourglass" mouse pointer never goes away. The program does not stop until you press the **Stop** button, because it is always looking for new events.

To stop the search for new events, press the **Stop** button. After you have done this, you can enter new IDs (org/host ID pairs, or org/host/app ID tuples) and restart the search with the **Restart** button. You can also use the various save and print utilities to store the data you have collected.

Press **Stop** and then **Close** to stop the event search and close the window.

The events are displayed with an OpenView/NetView utility called **xnmappmon**. You can change the fonts and layout of this utility by changing the application defaults file for this utility (see your network management platform documentation for details).

Remotely Configuring NetRanger Daemons

There are two ways to change the configuration of NetRanger daemons running on remote NSX boxes.

You can use the **nrget/nrset** infrastructure to modify daemon characteristics based on "tokens" that are specified by the user. These commands can be run at the command line of the Director machine.

You can also use the graphical browser utility that comes with the Director. To bring up the utility, click on either an Application or a Machine symbol, and then select the menu option **Security..Configure.**

Finding the Version of a Remote NetRanger Daemon

To determine the version of a particular application on a remote machine, simply click on the application icon, and then select the **Security..Version** menu option.

Changing Map Configuration Parameters

There are three global Map-level configuration parameters that can be set. To see these parameters, on HP systems, select the menu option **Map..Maps..Describe/Modify.** On IBM systems, select the menu option **File..Describe Map.** You will then see a pop-up window. On this window, choose the **NetRanger/Director** application, and then press the **Configure** button.

A window with three parameters will appear. You will see the following questions:

1. **Default lowest event severity that generates marginal icon?**
2. **Default lowest event severity that generates critical icon?**
3. **Should new security alarms be shown on the IP Map?**

The answer to the first question specifies the minimum severity an event must have before a marginal (yellow) Alarm is generated. For instance, if you set this value to 2, then if any new applications are created, these applications will have marginal alarms generated for events whose status is two and higher. Of course, if you manually reconfigure the Application symbol to have a new marginal status threshold, then this default value will be overridden.

The answer to the second question specifies the minimum severity an event must have before a critical (red) Alarm is generated. For instance, if you set this value to 3, then if any new applications are created, these applications will have critical alarms generated for events whose status is three and higher. Of course, if you manually reconfigure the Application symbol to have a new critical status threshold, then this default value will be overridden.

The third question asks if you want alarm icons to be drawn on the IPMap. *This option has no effect in the Director 1.0. It will take effect for versions 2.0 and beyond.* The IPMap is the submap hierarchy created by the ipmap application. The ipmap application is the part of OpenView/NetView that draws a picture of the IP Topology. The advantage of having alarms represented on the IPMap is that you can view Fault status and Security status on the same screen. The disadvantage is that performance is degraded because extra icons are being created.

Changing IP Addresses and Hostnames

If you change the IP characteristics of either a Director or NSX box, or if you change the NetRanger communication infrastructure characteristics (like hostId, orgId, host name, and organization name), you *must* ensure that the appropriate configuration files have been changed on *all* your NetRanger machines, including the Director machine.

If an IP Address is changed, ensure that the /usr/nr/etc/routes, /usr/nr/etc/sensord.conf, and /usr/nr/etc/managed.conf files are changed as necessary. If host or organization names have changed, ensure that the /usr/nr/etc/auths, /usr/nr/etc/destinations, /usr/nr/etc/hosts, /usr/nr/etc/routes, and /usr/nr/etc/smid.conf reflect the new configuration. If host or organization IDs have changed, ensure that the /usr/nr/etc/hosts has changed.

If the IP address or hostname of the network management station must be changed, consult your network management documentation to learn about what configuration changes must be made to your network management platform. On HP systems, it is recommended that you shut down the user interface, stop the OpenView daemons, stop the NetRanger daemons, and then use SAM to reconfigure the IP information. If you are changing the hostname, you should run `/etc/set_parms hostname` to ensure that the Common Desktop Environment is aware of the new hostname. Once this is done, and once any additional OpenView-specific configuration is complete (as specified in the OpenView documentation), it is recommended that you reboot your machine.

Changing Registration Files

All OVw Applications have a configuration file called a Registration File that tells the User Interface (OVw) how to treat the application. On HP systems, registration files are kept in `$OV_REGISTRATION`, and on IBM systems, the files are stored in /usr/OV/registration.

In general, registration files should not be modified, but there are a few circumstances in which it is helpful to edit registration files. Registration files contain the command that is actually used to launch the OVw Application, so registration files are good places to edit an OVw Application's command-line parameters.

There are four *nrdimap* command line parameters that you may need to edit. The four parameters are given below:

Option	Parameters	Function
-t	<none>	turns debug Tracing on
-p	<none>	Propagates alarms to ipmap
-m	non-zero int.	default marginal status threshold
-c	non-zero int.	default critical status threshold

If you are ever asked to enable tracing for debugging purposes, add a " -t" after word "nrdirmap on the line that looks like this:

```
Command -Shared -Initial "nrdirmap";
```

If you want alarms to be propagated to the ipmap so the nodes in the ipmap display security status as well as fault status, add the "-p" option.

By default, an event has to have a severity of 3 to have a marginal icon created, and a severity of 4 to have a critical icon created. To change these values, use the "-m" and "-c" options. For instance, to change the default values to be 2 and 3, respectively, change the line to look like this:

```
Command -Shared -Initial "nrdirmap -m 2 -c 3";
```

Changing the Number of Events Displayed in Event List

When you select a Machine or Application symbol and select the menu option **Security..Show Current Events**, by default, the last 100 events associated with that entity are displayed (if less than 100 events are known, then all of the events are displayed). To change the number of events that are displayed, use an editor to modify the nrdirmap file, which is stored in $OV_REGISTRATION/C on HP systems and /usr/OV/registration/C on IBM systems.

Replace the "100" with the number of your choice on the line shown in bold.

```
Action show_events  {
        SelectionRule (isWheelGroup && (isMachine ||
isApplication));
        MinSelected  1;
        Command "sh -c 'unset OVwSessionLoc \;
                $OV_BIN/xnmappmon \
                -selectList \" $(OVwSelections) \" \
                -commandTitle \" Show Current Events for \"
\
                -appendSelectList \
                -appendSelectListToTitle \
                -multipleDialogs \
                -headingLine 2 \
                -geometry 900x600 \
                -followOutput \
                -unbufIO \
                -stopSignal 9 \
                -cmd /usr/nr/bin/filterLogByHostApp -l 100'
";
    }
```

Searching for Symbols

HP OpenView and IBM NetView for AIX provide fairly powerful search utilities. These search utilities can be used to locate symbols that match certain criteria. The following three search functions might be useful when searching for alarm symbols:

- Locate by Object Attribute
- Locate by Symbol Type
- Locate by Symbol Status

To use the Locate function, simply select **Locate..Objects** from the main menu, and then pick the type of search you want.

For example, to see how many unresolved **String Matches** you have, you could search by Symbol Type, and select the **Alarm:Content** symbol type. To determine how many critical elements you have in your network, you could do a search by Symbol Status, and then search for Critical (red) elements. Finally, to search for an alarm from a particular source IP address, you could search by Attribute, and then pick "Source IP Address" from the list of attributes, and then type in the source IP address you want to find.

Setting the Home Submap

When you start up the user interface, a submap that is designated the "home submap" is opened. By default, the root (top) level submap is the home submap. You may want to change the home submap to the child submap of the top level NSX Collection (the submap you get when you double-click on the NSX Collection that appears on the root level submap). If you want to make this change, do the following:

1. **Double-click the** NetRanger **icon.**
2. **Select the menu option** Map..Submaps..Set This Submap As Home.

Changing a Submap Background

It is sometimes helpful to place a background picture on a submap to help identify the submap quickly. Submap backgrounds can also be used to help provide context for the different symbols on the submap (for instance, machine icons positioned strategically on a picture of a floor plan could help mark where the machines reside physically).

To add a submap background, open the submap that you want to change, and then select the menu option **Map..Submaps..Describe/Modify.** Under the **Background Graphics:** heading, press the **Browse...** button. From the pop-up list, select the background graphic of your choice. "usastates.gif" is a popular choice. You could also create a custom GIF file with any graphics program, and use that GIF file as an OpenView submap background. Press **OK** and then press **OK** again.

Repositioning Symbols on a Submap

You can use the mouse pointer to move symbols to different positions on a submap. However, if symbols are added to or removed from the submap, the user interface will automatically reposition *all* of the symbols on the submap, and the customization will be lost.

To prevent this, it is usually best to turn **automatic layout** off. To do this, select **View..Automatic Layout**, and select the **off** option for either the current submap (if you are only repositioning symbols on a small subset of submaps) or for all submaps (if you reposition symbols frequently).

Hiding Symbols

Under some circumstances, you might want to prevent a symbol from appearing on a given submap, but you might not want, or be able, to delete the symbol. For instance, there could be a machine in a collection that you don't care about, but you can't delete it because it has unresolved alarms. Assume for the moment that there is some reason why you don't want to delete the alarms. In a situation like this, it is best to *hide* the symbols.

To hide a symbol, select it, then choose the menu option **Edit..Hide Object(s)**. You are given a choice of hiding **This Submap** or **All Submaps**. Pick the option of your choice.

To "unhide" a symbol, simply select the **Edit..Show Hidden Objects** menu function.

Changing the Status Propagation Schemes

When a symbol has Compound Status Source, the status (color) of the symbol is based solely on the status of the symbol(s) in that symbol's child submap. HP OpenView and IBM NetView for AIX provide the user with user-selectable sets of "rules" that the User Interface uses to determine the status of a symbol based on the status of the symbols in the child submap. These "rulesets" are called **Compound Status Source Propagation Schemes**, and the ruleset you choose will affect the color of the icons on the map.

To change the status propagation scheme, Select the menu option **Map..Maps..Describe/Modify**, select one of the radio buttons associated with a scheme, and then press **OK**.

WheelGroup Corporation highly recommends that you select the Propagate Most Critical scheme.

Consult the documentation provided with your network management platform for more information about Compound Status Source.

Changing Appearances, Fonts, Window Sizes, Colors, Etc.

In HP OpenView and IBM NetView for AIX there are special ASCII files called **Application Default** files that contain parameters that can be customized to change the look and feel of certain applications. To change fonts, window sizes, colors, etc. for the user interface in general, edit the OVw file which is in `$APP_DEFS` on HP systems and /usr/OV/app-defaults on IBM systems.

To modify the attributes of the various XNm applications, modify the XNm* files. The **Show Current Events** window uses an application called **xnmappmon** (X-Node Manager Application Monitor) to display data. If you want to change the appearance of this window, make the necessary modifications to this file.

eventd

Overview

eventd is a daemon service shipped with the NetRanger Director package that allows users to receive notifications of alarm events via e-mail. *eventd* receives copies of alarms from *smid*; arranges them into a readable format, and e-mails them to users based on information stored in configuration files. An example e-mail follows:

```
From netrangr@director1.wheelgroup.com Tue Nov  5 12:01:10 1996
From: netrangr@director1.wheelgroup.com
Date: Tue, 5 Nov 1996 12:07:42 -0600
Subject: Alarm OUT->IN Level 2.

Date: 1996/11/05,12:06:00
NSX: [10001.1.100]
Attack: 8000,502 Level: 2
Addrs: 192.216.46.55:80->207.18.164.150:40134
Names: webcrawler.com->lookout.wheelgroup.com
Sig: 8000 "String Match"
Msg: "golf"
```

Basic Setup

eventd is not configured when the package is installed. To set up *eventd*, do the following:

- set up event script's configuration file
- set up eventd configuration file
- set up smid configuration file
- set up eventd to start

Set Up the Event Scripts' Configuration File

The event script converts alarms into e-mail messages and sends them to a configurable destination. *eventd* must be configured before it can send alarms. The default configuration file is a template with sample comments only. Unlike the configuration files for Netranger's daemon services, the configuration files for the event *script* are located in /usr/nr/bin/eventd. Do the following steps to configure the event script:

1. **Add your organization definition.**

   ```
   ORGANIZATION    100
   ```

2. **Add users to receive Type 2 alarms. Only one line for Type 2 alarms is accepted.**

   ```
   2       netrangr
   ```

3. **Add users to receive Type 3 alarms. Only one line for Type 3 alarms is accepted.**

   ```
   3       netrangr
   ```

4. **Add users to receive Type 4 alarms. Multiple lines may be added for Type 4 alarms. Type 4 alarms have 5 levels, a source, a destination, and recipients.**

   ```
   # Level 4 Alarms
   #
   # Source and Destination of Alarm
   # Src: OUT,IN,-
   # Dst: OUT,IN,-
   # "-" denotes either
   #
   #Type  #Level  #Src    #Dst    #Recipients
   4      1       OUT     IN      root
   4      2       OUT     IN      root
   4      3       OUT     IN      root
   4      4       -       -       user1
   4      4       OUT     IN      user2
   4      4       IN      OUT     user3
   4      4       -       IN      user4
   4      4       -       OUT     user5
   4      4       IN      -       user6
   4      4       OUT     -       user7
   4      5       -       -       root,user1,user2
   # Multiple lines are allowed for type 4 alarms.
   4      5       -       -       mcnealy@sun.com,bgates@ms.com
   ```

NOTE

Level denotes the level of the alarm you wish to select. Source and Destination have three possible settings: "IN," "OUT," and "-". IN denotes that the address must be inside the network. OUT denotes that the address must be from inside the network. The dash "-" symbol denotes that the address can be either. Source denotes the location of the IP address you wish to select for the recipients (as detailed above). Destination denotes the location of the IP address you wish to select for the recipients (as detailed above). Recipients denotes the mail destination of events that match the preceeding selection process.

Set Up eventd's Configuration File

Before *eventd* can send alarm levels to destination, the following line in /usr/nr/etc/eventd.conf must be added:

```
EventApplication <UniqueID> <lowestLevelToSend> <locationOfScriptRelativeTo/usr/nr/bin>
```

The following example defines one script for one organization:

```
EventApplication 1002 2 ./eventd/event
```

Set Up smid's Configuration File

In order for eventd to receive alarm events from *smid* the configuration file smid.conf must contain a `DupDestination` entry for *eventd*. An example entry might look as follows:

```
DupDestination director1.wheelgroup eventd  2 ERRORS,COMMANDS,EVENTS
```

Set Up eventd to Start

Edit /usr/nr/etc/daemons by uncommenting the following line:

```
# nr.eventd
```

NOTE

If the above line is uncommented, then eventd will not be able to start after a reboot.

Advanced Setup

Monitoring Multiple Organizations.

If you are monitoring only one (1) organization, use event.conf as your configuration file. Otherwise, build a seperate copy of event and event.conf for each organization. To do this, add links pointing to the original event under the name of the organization and make a copy of event.conf under the same name as the new event script. To edit each organization-specific configuration file, follow these steps.

1. **Create a script file with the same name as the configuration file. Use UNIX link command:**

   ```
   %cd /usr/nr/bin/eventd

   %ln -s ./event ./event_wheelgroup
   ```

   The script file, when run, will look for a configuration file by the same name with ".conf" appended. Use event.conf as an example.

2. **Make a copy and edit it according to your local needs:**

   ```
   %cp event.conf event_wheelgroup.conf

   %vi event_wheelgroup.conf
   ```

3. **Edit /usr/nr/etc/eventd.conf by adding organization designations, for example:**

   ```
   EventApplication 1002 2 ./eventd/event_wheelgroup

   EventApplication 1003 2 ./eventd/event_organization2
   ```

   In the example above, two scripts for two organizations are being added. The organizations are designated as wheelgroup and organization2.

For example, a new script for wheelgroup is made under the name of event_wheelgroup, and event_wheelgroup.conf, respectively.

The lines look this before configuration:

```
-rwxr-x---   1 netrangr netrangr    6603 Nov 22 17:19 event
-rwxr-x---   1 netrangr netrangr    1021 Nov 22 15:30
event.conf
```

and like this after configuration:

```
-rwxr-x---   1 netrangr netrangr   6603 Nov 22 17:19 event
-rwxr-x---   1 netrangr netrangr   1021 Nov 22 15:30 event.conf
lrwxr-xr-x   1 netrangr netrangr      5 Nov 22 15:31 event_wheelgroup -> event
-rwxr-x---   1 netrangr netrangr   1305 Nov 23 12:47 event_wheelgroup.conf
lrwxr-xr-x   1 netrangr netrangr      5 Nov 22 15:31 event_organization2 ->
event
-rwxr-x---   1 netrangr netrangr   1305 Nov 23 12:47 event_organization2.conf
```

Changing the Event Error Notification user

Event Error Notification is in the event script. It looks for its own configuration file, and if it cannot find it, or it has an error processing the alarm, it notifies the person(s) named as MAIL_FAILURE. To change user to receive e-mail on problems with the configuration file, edit eventd in the following manner:

```
%vi event
MAIL_FAILURE="userToReceiveEmailOnFailure@somehost.com,netrangr
"
```

For example:

```
MAIL_FAILURE="bob@alarmsRus.com,postmaster@walrus"
```

In the above example, "postmaster@walrus" is the second user added to the MAIL_FAILURE line.

Overview

nrConfigure is a Java-based graphical user interface (GUI) that allows you to remotely configure NetRanger applications and access those configurations. It supports all the functionality of the individual nrget, nrgetbulk, nrset, nrunset, and nrexec commands. This GUI-based environment allows you to see an application's tokens, each token's actions, and each action's optional values. Figure xx illustrates the nrConfigure window.

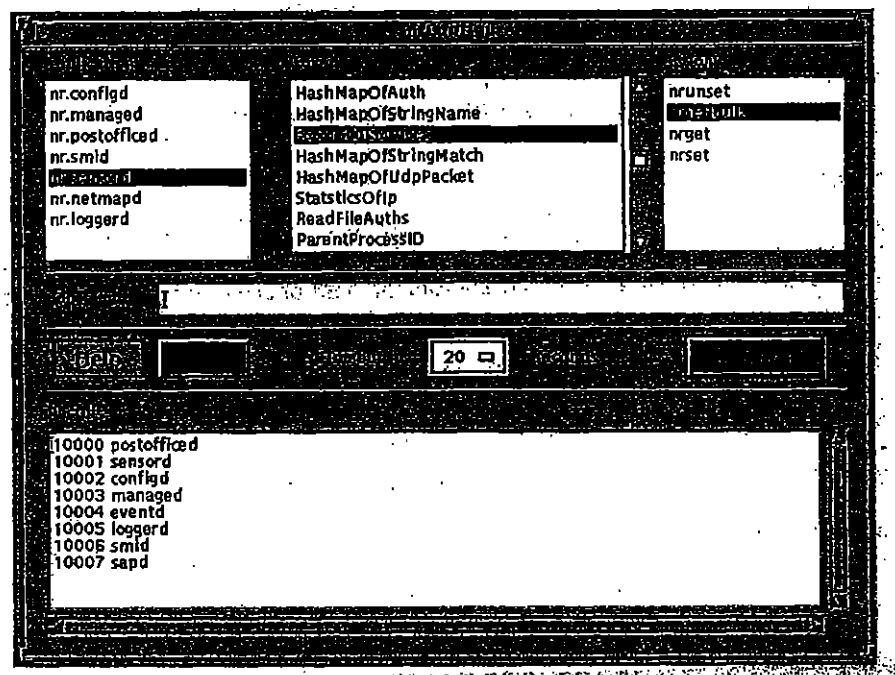


Figure IV-11: The nrConfigure GUI

Architecture

The nrConfigure architecture supports communication GUI works in the following manner. When you press **Execute** for a given application, token, and action, a packet is sent to *configd,* which sends it on to *postofficed. postofficed* sends it on to the proper daemon, which processes the requests and replies. The reply returns through *postofficed* to *configd.* The reply appears in the nrConfigure's Results window.

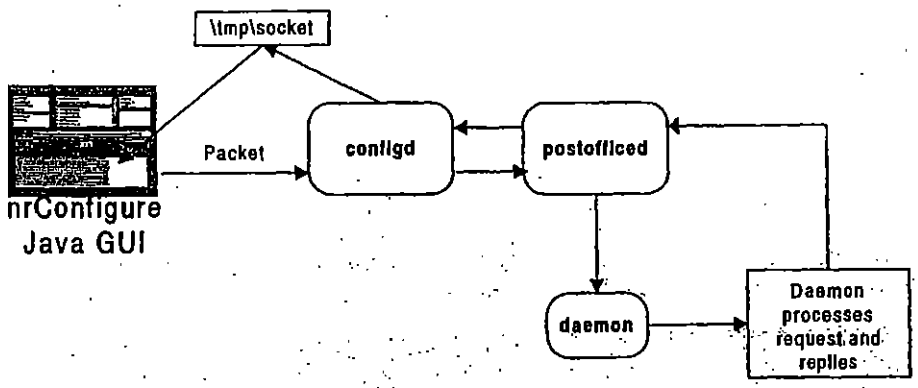


Figure IV-12: nrConfigure Architecture

Starting nrConfigure

nrConfigure is designed to configure daemon applications on an NSX or Director system. This is done by either selecting a machine icon from an OpenView security map or by invoking nrConfigure directly from the command line with the NSX or Director's organization and host id.

To start nrConfigure from within the Director:

⇒ select **Configure** from the **Security** menu.

To start nrConfigure from the NSX command line:

⇒ enter this command:

```
/usr/nr/bin/nrConfigure <Organization ID>.<Host ID>:<Host Name> &
```

NOTE

The `<Organization ID>` and the `<Host ID>` can be found in /usr/nr/etc/hosts.

NOTE

You may run multiple copies of nrConfigure at the same time.

Quiting nrConfigure

To quit any nrConfigure GUI, simply press the cancel button. Pressing the cancel button on one GUI will not quit any other nrConfigure GUI session.

Help with nrConfigure

If you need help with nrConfigure, do the following steps:

1. **On the nrConfigure GUI, select an Application.**
2. **Press the Help Button.**

   A help window is displayed for the selected application. Help is also listed for all of the application-supported tokens.

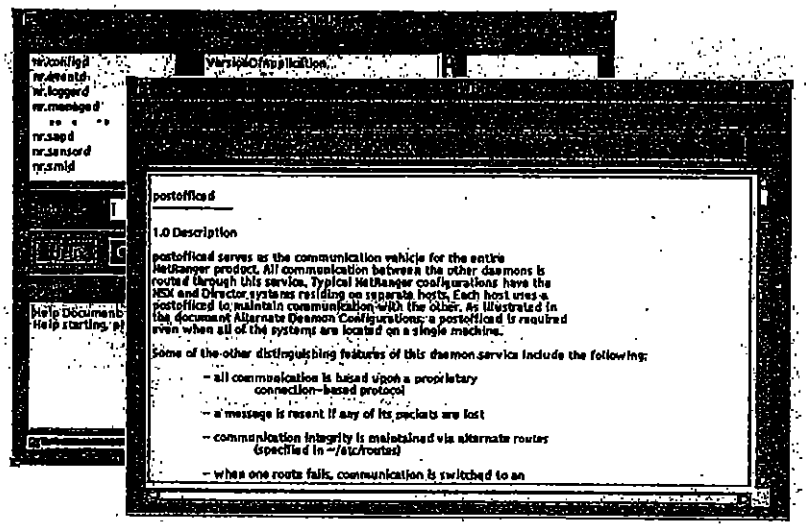


Figure IV-13: nrConfigure Help

Configuring an Application with nrConfigure

To configure an applicaiton with nrConfigure, follow these steps:

1. **Select the Application you wish to configure in the Application selection box.**

   The GUI will display that application's allowed Tokens.

2. **Select the Token you wish to execute in the Token Selection box.**

   The GUI will display that token's allowed actions.

3. **Select the Action you wish to execute in the Action selection box.**

   The GUI will display the following information:

   - Any Optional Parameters, if required
   - Any default values for Optional Parameters
   - Any set values for Optional Parameters

NOTE
The GUI will display Parameters without current values as the parameter's name in "<>" brackets.

4. **Press the Execute button.**

   The results will display in the Results field.

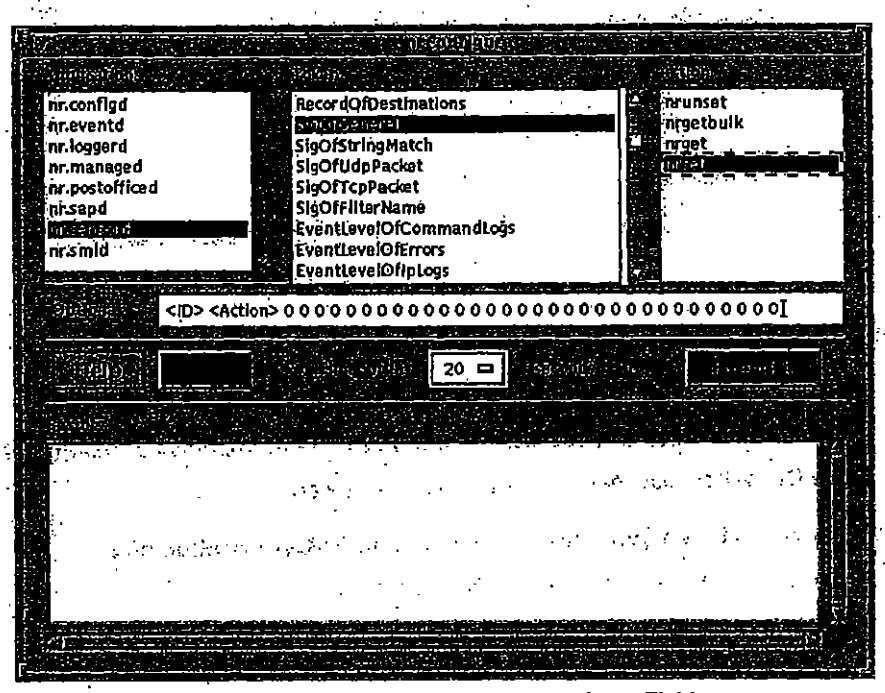


Figure IV-14: Example of Parameters in Optional Field

V Using NetRanger with RDBMS

Working with NetRanger's Data Management and Archiving Feature

NetRanger's Security Analysis Package (SAP) is a set of data analysis tools that analyze NSX data. The SAP consists of two components: database export utilities to relational databases and one or more data analysis tools. NSX data collection serves as the foundation for the SAP subsystem, and is based on the *loggerd and sapd* services diagrammed in Figure V-1.

Writing to intermediate flat files in this manner provides levels of **fault tolerance** and **performance** that cannot be achieved by writing directly to a database. While the SAP currently ships with *sapd* drivers only for **Oracle** and **Remedy**, it can also be configured to write to other databases, such as Sybase and Informix. Example scripts are shipped with the Director software that show how a database's native bulk load tools can be easily integrated with *sapd*.

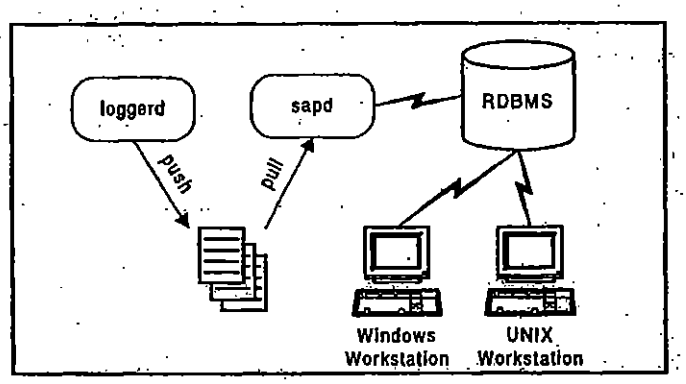


Figure V-1: The NetRanger Data Archival Process

The Push-Pull Model for Data Archiving

NetRanger's data archiving function is performed by *loggerd* and *sapd*. These daemons use a simple **push-pull** mechanism to migrate data into a remote database. *loggerd* receives event notification from the NSX and Director daemons and "pushes" log data into flat files, which are serialized based on a configurable size or time interval. *loggerd* writes its current log file into /usr/nr/var and uses the following file-naming convention:

```
log.DATETIME
```

Where `DATETIME` is 199611050915 (Year,Month,Day,Hour,Minute).

sapd, which has its own polling interval, "pulls" the data from *loggerd* and executes the user-defined database load procedure. The load procedure consists of control scripts and a loader program. *sapx* is the default loader program that loads data from the log files into Oracle or Remedy.

Log File Switching/Serialization

Based on user-configurable parameters, *loggerd* pushes its current log file into /usr/nr/var/new and opens a new one in /usr/nr/var. This process is called **serialization** and is part of the log file-switching paradigm.

These tokens define the file location for *loggerd's* serialization:

- **FilenameOfLog**—path to current log file (`.DATETIME` is automatically appended)
- **FilenameOfNew**—path to pushed holding area

The log file growth threshold is defined in terms of file size or time intervals. When either of these conditions are met, the current log file is closed, pushed to the /usr/nr/var/new directory, and a new log file is opened in /usr/nr/var.

These tokens are used to set the maximum time and file size limits for the log file serialization process:

- **NumberOfSwitchMinutes**—sets the maximum time the current log file is allowed to stay open until it is serialized.
- **NumberOfSwitchBytes**—sets the maximum file size the current log file is allowed to grow to until it is serialized.
- **NumberOfSwitchMinutes** and **NumberOfSwitchBytes**—work in conjunction for setting a ceiling threshold for switching and serialization (either when the elapsed time expires, or when the file grows to be a certain size).
- **SwitchFile**—executes a switch/serialization on-demand.

Bypass Same Minute Log File Switch

loggerd has an automatic feature that forces a minimum switch time of one minute (except for the start-stop handling described below). This means that, even if the size threshold is triggered, *loggerd* will not serialize a log file until it has been open for at least 60 seconds. This feature eliminates runaway log file serialization when the network is experiencing heavy traffic.

Non-Clobber Sub-Serialization

loggerd also includes a feature prevents log files from being "clobbered" (overwritten) when the NetRanger daemons are started and stopped several times in a minute. This sub-serialization is automatic and moves log files with similar names into the /usr/nr/var/new area and gives them a distinctive file extension.

Examples of Automatic Serialization

```
NumberOfSwitchMinutes = 60

NumberOfSwitchBytes = 100000
```

This means that either every hour, or when the file size reaches 100K, the current file will be closed and moved to the /usr/nr/var/new area. A new file is then opened immediately in the /usr/nr/var area.

The following examples show a modified ls -l listing of the /usr/nr/var/new directory. *loggerd* was running with the switch token values listed above.

Example #1 (Ceiling Threshold)

In the following example, the files that were smaller than 100K were switched every hour; otherwise they were switched when the size reached 100K.

```
FILESIZE  DATE   TIME  FILENAME
   61058 Nov  7 06:19 log.199611070519
   60524 Nov  7 07:19 log.199611070619
   70008 Nov  7 08:19 log.199611070719
  103730 Nov  7 08:40 log.199611070819
   76926 Nov  7 09:40 log.199611070840
   83476 Nov  7 10:40 log.199611070940
  100101 Nov  7 10:48 log.199611071040
   92029 Nov  7 11:48 log.199611071048
   90902 Nov  7 12:48 log.199611071148
   95840 Nov  7 13:48 log.199611071248
  101681 Nov  7 14:20 log.199611071348
  100019 Nov  7 14:51 log.199611071420
  104448 Nov  7 15:24 log.199611071451
  103033 Nov  7 15:50 log.199611071524
  109992 Nov  7 16:08 log.199611071550
```

Example #2 (Bypass Same Minute Switch)

In the following example, very heavy traffic occurred between 10:14 and 10:16. The log file which was opened at 10:14 grew to 259K within a minute and was then switched.

```
FILESIZE  DATE   TIME  FILENAME
  100585 Nov 11 09:16 log.199611110849
  100358 Nov 11 09:49 log.199611110916
  101432 Nov 11 10:14 log.199611110949
  259555 Nov 11 10:15 log.199611111014
  177152 Nov 11 10:16 log.199611111015
  100307 Nov 11 10:44 log.199611111016
  100333 Nov 11 11:32 log.199611111100
  100065 Nov 11 11:56 log.199611111132
  100049 Nov 11 12:26 log.199611111156
```

Example #3 (Sub-Serialization)

In this example, nrstart and nrstop were executed as quickly as possible and the NetRanger daemons started and stopped 6 times in a minute (16:44). Notice how the .## is appended to the file name so that previous files are not overwritten.

```
FILESIZE  DATE   TIME  FILENAME
     178 Nov  8 16:44 log.199611081644
     313 Nov  8 16:44 log.199611081644.1
     308 Nov  8 16:44 log.199611081644.2
       0 Nov  8 16:44 log.199611081644.3
     306 Nov  8 16:44 log.199611081644.4
     178 Nov  8 16:44 log.199611081644.5
```


sapd works with *loggerd's* pushed output log files, pulls them from the /usr/nr/var/new area, and loads them into your database.

The database destination is fully user-configurable. You can use the default methods provided with the *sapd* package, or refine them to meet the special needs of your environment.

Oracle is the native database for *sapd.* Full support is provided in the SAP package. Example templates are also provided for loading into other DBMS products.

sapd Processing Model

sapd is always running and checks the /usr/nr/var/new area for any available files every 60 seconds. When it finds a new file, it launches a processing cycle. This cycle is illustrated in Figure V-2.

You can delay the loading once a logfile is found by setting the MinIdleTime token to an integer > 60 (seconds). This feature is included to slow down the *sapd's* launch of the processing cycle.

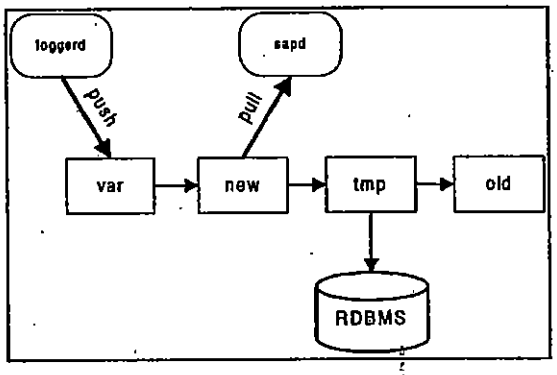


Figure V-2: SAP Log File Flow

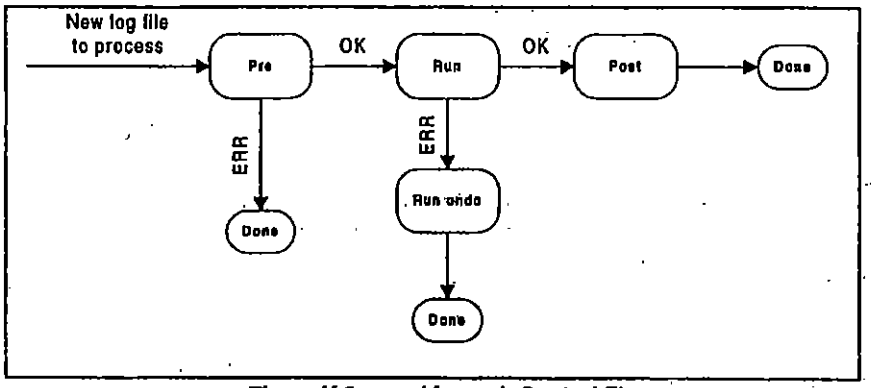


Figure V-3: sapd Launch Control Flow

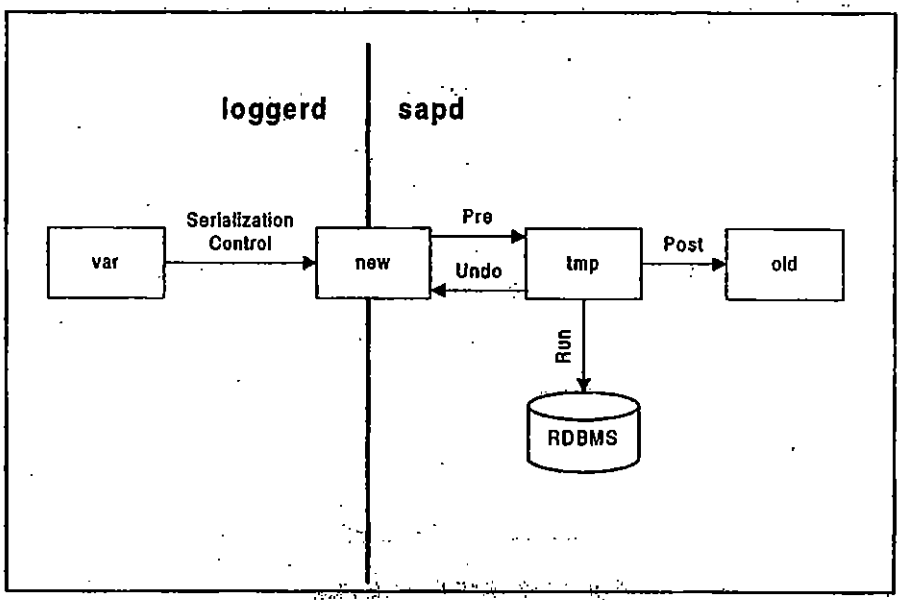


Figure V-4: SAP Data and Control Flow

nr.sapd always calls its scripts with a single argument: the filename of the current log file (name only, not the full path). nr.sapd also populates environment variables for each called script. The variables reflect the current token values.

This launched script (or child) uses the values of the variables in its operations. Refer to the following list to see how the token values map into the environment variables:

sapd token	child environment var
LogFilePathNew	PATH_NEW
LogFilePathTmp	PATH_TMP
LogFilePathOld	PATH_OLD
DBUser	SAP_DB_USER
DBPass	SAP_DB_PASS
DBUser2	SAP_DB2_USER
DBPass2	SAP_DB2_PASS
DBAux1	SAP_DB2_LANG
DBAux2	SAP_DB2_SERVER
DBAux3	SAP_DB2_SCHEMA
/usr/nr/bin/sap	SAP_BIN

sapd looks at /usr/nr/var/new for any files that are pushed into this area.

When it finds a file, it starts the processing cycle with PRE.

PRE moves the file from /usr/nr/var/new to /usr/nr/var/tmp. if PRE was successful, RUN is launched. If PRE was not successful, nothing more is done.

RUN takes the file in /usr/nr/var/tmp, manipulates it, and calls a program to load it into your destination

If RUN was successful, POST is launched. If RUN was not successful, RUN_UNDO is launched.

POST cleans the files in /usr/nr/var/tmp and moves the original logfile into /usr/nr/var/old.

RUN_UNDO restores to the state before PRE was launched. It moves the original log file from /usr/nr/var/tmp back into the /usr/nr/var/new area.

Data and Control flow can be summarized as follows:

PRE->RUN->POST (no errors, log file is loaded and moved to old)

or

PRE->RUN->RUN_UNDO (error occurred in RUN, revert to log file state before PRE)

Token Initialization

The defaults for the tokens are as follows:

LogFilePathNew /usr/nr/var/new

LogFilePathTmp /usr/nr/var/tmp

LogFilePathOld /usr/nr/var/old

LogFileMask log.

(Using these defaults is recommended, but you have the flexibility to change if necessary.)

ControlPre /usr/nr/bin/sap/load_pre.sh

ControlRun /usr/nr/bin/sap/load_run.sh

ControlPost /usr/nr/bin/sap/load_post:sh

ControlRunUndo /usr/nr/bin/sap/load_run_undo.sh

(These tokens can be changed, either permanently or on-the-fly to facilitate custom operations. See documentation for *configd* and Java configuration for information about token operation.)

Tokens for Oracle:

DBUser : username for oracle access

DBPass : password that goes with DBUser

These must be set before you can connect to Oracle. They are set to match whatever user/pass was specified in the DBMS setup (create user).

Tokens for Remedy or other DBMS:

DBUser2 : username

DBPass2 : password that goes with DBUser2

DBAux1 : auxiliary environment information

DBAux2 : auxiliary environment information

DBAux3 : auxiliary environment information

Using Default Scripts and sapx

The *sapd* package includes default scripts and a loader program called sapx. sapx is used to load various files into Oracle or Remedy.

sapx supports the following input/output types:

Into Oracle: logs: error, command, alarm

config files: signatures

Into Remedy: logs: alarm

The default scripts are located in /usr/nr/bin/sap

load_pre.sh moves file from /usr/nr/var/new to /usr/nr/var/tmp

load_run.sh splits file into sub-files grouped by log type, calls sapx to load into oracle

load_post.sh moves original log file from /usr/nr/var/tmp to /usr/nr/var/old

load_run_undo.sh restores previous state before processing began:

moves original log file from /usr/nr/var/tmp to /usr/nr/var/new

sapx Command Line Parameters

sapx requires the following three command line parameters:

LogFileName—the path and filename to the log file to be processed

SourceType—a number (2, 3, 4, or 5) designating the log file type (2=errors, 3=commands, 4=events/alarms, 5=signatures)

TargetType—a number (1 or 3) designating what the type of output to load into (1 is Oracle, 3 is Remedy). Use the /usr/nr/bin/sap/skel/populate_sigs.sh. Type 5 loads of the /usr/nr/etc/signatures needs to be performed only on initialization and when that file changes.

sapx Messages

The default load_run.sh script redirects sapx messages to /usr/nr/var/messages.sapd. Review this file to get an idea of what has been loaded. This file is useful for troubleshooting when no database records are being inserted.

These two lines are examples of sapx messages:

```
sapx:(/usr/nr/var/tmp/log.199611050003.split.3,3,1) Ora(user@ora1) recs(2, 2, 0, 0)

sapx:(/usr/nr/var/tmp/log.199611050003.split.4,4,1) Ora(user@ora1) recs(862, 862, 0, 0)
```

The first group is the input parameters to sapx. The second group is the output destination. The third group describes the records processed:

recs(READ , DB INSERTED , Blank , Skipped)

When loading the signatures file, you will see a records message that looks something like this:

```
recs(66, 56, 3, 7)
```

This means that 66 input lines were read from the input file, 56 were inserted into the database, 3 lines were blank and ignored, and 7 lines were ignored for some other reason (i.e., comments in the file).

Use this instrumentation from sapx to monitor success and failures in your database load process.

Changing Scripts to Deselect Log Type(s)

sapd, by default, will attempt to load all three log types:

(2) errors, (3) commands, and (4) alarms

In /usr/nr/bin/sap/load_run.sh, change the line for filetype in 2 3 4, removing the numbers corresponding to the log types you wish to ignore.

For example, if you only want to load alarms, and skip errors/commands,change the 'for' line to the following:

```
for filetype in 4
```

Changing Scripts to Activate Remedy ARS

Replace control script, /usr/nr/bin/sap/load_run.sh with /usr/nr/bin/sap/skel/load_run.sh.remedy. Notice how the extra environment variables are set before calling sapx.

Alternate Loading Procedures

Change the Schema

If you want to load the data into a schema different that the default, or you are using a non-Oracle database, refer to the SQL*Loader control file in /usr/nr/skel/sap. This is a sample template file which is used by Oracle's SQL*Loader. Other major databases have similar Loader programs which work with similar template files.

Use the create_log*.sql files as a template for the schema declaration.

Use the oraldr_log*.ctl files as a template for SQL*Loader field mappings.

Use the load_run.sh.oraldr as an example load_run.sh control script.

Modifying the Scripts

Be careful with changes and make sure you understand the processing data flow and control flow models before attempting customizations. As a standard practice in good development, rigorous testing should be performed before a program/script is put into an operational environment.

Control Script Exit Status

The scripts must return an exit status so that *sapd* can control the flow and recover from errored runs. If the script completed without error, return 0

If an error was encountered during processing, return 1 or greater

Notes on Shell Script Exit Codes

The SH variable $? is updated with exit status after each command is executed. The command, exit, leaves the script and returns status to its invoker.

exit called without a number returns the $? of the last command executed

exit called with a number, (e.g. exit 2) returns that number to the invoker as the return status

If a script does not call exit after the last line in the script, an implicit exit is called with the status of the previously executed command.

Null Control Scripts

if the Control token for the script is blank, or the script named by the token is not found or not executable, the state skip feature is invoked. This feature skips to the next state in the control flow model, returning if the null script is an endpoint (POST or RUN_UNDO). This is primarily an error handling strategy, but can be utilized if the process of customizing your loader scripts.

Switching Control Scripts

You can have separate load_pre.sh, load_run.sh, load_post.sh scripts for custom applications. You can change these on the fly with the Control Pre, Control Run, and Control Post tokens.

Usage Recommendations

Loading large log files (1MByte or more) can be challenging for a small database when full Rollback support is enabled. Therefore, we recommend that you try to keep the log files small. The exact setting for NumberOfSwitchBytes depends on how fast your log files grow. This is a function of your environment, configuration, and network activity.

sapx has an automatic feature that executes intermediate commits every 500 records when a large file is loaded. This sapx feature effectively solves the large log file problem. You will need to take this situation into consideration when using a non-sapx loader program.

We also recommend using the default paths for FilenameOfLog and FilenameOfNew. You may want to configure the Director machine to have a separate large partition for /usr/nr/var or /usr/nr/var/new if you are having disk space problems.

Appendix A Troubleshooting

This section provides a series of troubleshooting procedures to follow in response to specific problems you may encounter when operating the NetRanger System.

Troubleshooting Procedures

Symptom

Unable to start or stop the NetRanger daemon processes when running nrstart or nrstop.

Possible Cause(s)

The most likely cause for this problem is that you are trying to run these utilities from a user account that does have access rights to the NetRanger daemons.

Possible Solution(s)

Make sure you are logged onto the NSX or Director platforms under the same account that was used to originally start its daemon services.

Symptom

The following error message is displayed when trying to start the Director system from an HP Terminal session:

```
Cannot write message to Director, errno = 2
```

Possible Cause(s)

This error occurs when the *smid* daemon is trying to write to a socket that does not exist. This may occur either because:

1. The Director's *nrdirmap* daemon was unable to create the socket because the OpenView user interface (OVW) has not been started.
2. The /usr/nr/etc/smid.conf file contains an entry for SocketNameOfDirector that differs from what nrdirmap uses.

Possible Solution(s)

Bring up OpenView before starting the Director.

Make sure that the SocketNameOfDirector is set to /tmp/socket.director.

Symptom

The following error message is displayed when trying to start the Director system from an HP Terminal session:

```
Cannot write message to Director, errno = 233
```

Possible Cause(s)

This error message is generated when the smid daemon is writing to a socket whose buffer is overflowing. This can occur for one of two reasons:

1. The Director's *nrdirmap* daemon is not running. Therefore, the /tmp/socket.director is overflowing.
2. *smid* is writing to a socket different from the one *nrdirmap* is trying to read from.

Possible Solution(s)

1. Bring up OpenView before starting the Director.
2. Make sure that the SocketNameOfDirector is set to /tmp/socket.director.

Symptom

The following error message is displayed when trying to start the Director system from an HP Terminal session:

```
Cannot write message to Director, errno = 239
```

Possible Cause(s)

This error message occurs when the Director's nrdirmap and smid daemon processes do not have adequate permissions to talk to one another via the sockets in /usr/nr/tmp.

Possible Solution(s)

The usual cause for this problem is that the Director's nrdirmap daemon was brought up under a user account that has different access rights than the user account used to start the background NSX-based daemons (that include smid). The solution to this problem is to stop both sets of processes and then restart them as user netrangr.

Symptom

The Director platform is unable to log or display the information from a specific NSX system.

Possible Cause(s)

Improper configuration of the NSX or Director /usr/nr/etc files.

Possible Solution(s)

1. The IP Address for the Director platform has been improperly specified in the NSX system's /usr/nr/etc/routes file. Make sure it is properly specified.
2. The Director's `<HostId>.<OrgId>` must be specified consistently across all of the NSX /usr/nr/etc files *auths*, *destinations*, and *hosts*. Make sure it matches across all of these files.
3. This problem is accompanied by the appearance of a /usr/nr/etc/errors.*postofficed* file that includes the following error message:

```
Net received message from unknown host <IP Addr xxx>:<Port No. yyy>.

Discarding message from <AppId>.<HostId>.<OrgId>
```

This usually means that the `<HostId>.<OrgId>` for the **NSX system** has been improperly specified **on the Director system** for the files enumerated in the preceding item.

Symptom

While the Director's security map contains an icon for an NSX, it fails to display any events for the NSX.

Possible Cause(s)

There are two possible causes for this behavior:

1. The Director *Severity Status* attributes are set too high to for the level of alarms being generated by the NSX.
2. The level of alarms generated by the NSX system to date fall below the routing threshold set in the NSX's /usr/nr/etc/destinations file.

Possible Solution(s)

1. From the Director interface, highlight the icon for the NSX system and then either press Ctrl+O or select Describe->Modify from the menu bar. Then select NetRanger/Director and press View/Modify. Make sure the *Minimum Marginal* and *Minimum Critical* status thresholds are low enough to register events from the NSX in question.
2. If the Director *Severity Status* thresholds are set properly, make sure that the routing threshold in the NSX's /usr/nr/etc/destinations file is set low enough to route information to the Director. In the following example, the *destinations* file has the event level set to '4', which means that only alarm events of level 4 and 5 will be forwarded onto the Director platform named *firefly.wheelgroup*:

```
1  sentinel.wheelgroup  loggerd   1 EVENTS,ERRORS,COMMANDS

2  firefly.wheelgroup   smid      4 EVENTS
```

Possible Solution(s)

The solution to this problem is to set the threshold to a lower level, such as 2.

Symptom

While the Director system is receiving an NSX's alarms properly, it fails to log or display any errors or commands that are known to have occurred on the NSX.

Possible Cause(s)

Incorrect configuration of /usr/nr/etc/destinations on the NSX system. Each system enumerates exactly what types of information should be routed to up to 32 different destinations.

Possible Solution(s)

Make sure that the destinations entry for your Director system includes entries for errors and commands. In the following example, one alarm event is being sent to the Director platform named firefly.wheelgroup. Its entries would have to match the entries for sentinel.wheelgroup in order for it to also receive errors and command records.

```
1  sentinel.wheelgroup  loggerd   1 EVENTS,ERRORS,COMMANDS
2  firefly.wheelgroup   smid      4 EVENTS
```

Symptom

While information is properly displayed when the Show Current Events utility is run from the Director, the mouse pointer turns into an hourglass and never changes back.

Possible Cause(s)

The mouse pointer remains an hourglass because current events utility continues to pull information from the Director log files as long as the window is up.

Possible Solution(s)

Press the Stop button to terminate the filtering application. You can then use the scrollbars and menu options. Press the Close button to close the display window.

Symptom

Even though an NSX's alarms are properly displayed in the Director security map, nothing is displayed in the Show Current Events window and the log files in the Director's /usr/nr/var directory all have a length of '0'.

Possible Cause(s)

The Director *loggerd* daemon is not listed as a destination in either the NSX's or the Director's configuration files.

Possible Solution(s)

There are two ways to solve this problem:

1. Create an entry in the NSX's /usr/nr/etc/destInations file for the Director's *loggerd, or*
2. Create a *DupDestInation* entry in the Director's /usr/nr/etc/smid.conf file to *redirect* event data to *loggerd* from *smid:*

```
DupDestination <HostId>.<OrgId> loggerd 1 EVENTS,ERRORS,COMMANDS
```

Where <HostId>.<OrgId> are the Director's values (e.g., *sentinel.wheelgroup*).

NOTE

Option #1 requires the NSX's *destination* file to contain ***two*** entries for the Director platform: one for *smid* and another for *loggerd*. This doubles the amount of network traffic that must be sent to the Director platform. While option #2 create some overhead for *smid*, it only requires ***one copy*** of event data to be sent over the communication pathway.

Symptom

The following error messages are generated simultaneously on one or more NSX and Director systems:

NSX:

```
Net received spoofed message, discarding message from <IP addr>, errno=0
```

Director:

```
Net received message from unknown host <IP addr>

Discarding message from <AppId>.<HostId>.<OrgId>, errno=0
```

Possible Cause(s)

One or more NSX and/or Director systems have an incorrect localhost entry in their respective /usr/nr/etc/hosts file.

Possible Solution(s)

Make sure that the localhost entries on each of your NetRanger systems' /usr/nr/etc/hosts file is correct. For example, if a machine has a <HostId> of 3 and an <OrgId> of 100, the localhost entry should match the <HostId>.<OrgId> for firefly.wheelgroup.

```
3.100 localhost


1.100 sentinel.wheelgroup

2.100 condor.wheelgroup

3.100 firefly.wheelgroup
```

Symptom(s)

Loss of connectivity to the BorderGuard router and the NSX Sensor, or no *managed* daemon

Possible Cause(s)

The NetRanger System is shunning itself.

Possible Solution(s)

If there is connectivity to the BorderGuard and the NSX Sensor, then

1. Restart *managed* on the NSX.
2. This will clear all shuns and will reinitialize *managed.*

If there is no connectivity to the BorderGuard and the NSX Sensor, then

1. Restart *managed* on the NSX and the BorderGuard.

Appendix B DBMS Requirements and Setup

In order for nr.sapd to stage data, you will need access to a database management system (DBMS). Oracle is the native database for the sap package. If you do not have Oracle, you can customize *sapd* to support non-Oracle databases, such as Sybase and Informix.

Oracle Install and Setup

Oracle activation for nr.sapd consists of the following steps:

1. **Plan your Oracle environment**
2. **Install/Mount the Oracle product**
3. **Set up environment variables**
4. **Verify generic connectivity**
5. **Create user**
6. **Verify NetRanger–Oracle connectivity**
7. **Create tables**
8. **Setup for Remedy ARS (Optional)**

Plan your Oracle Environment

Ask yourself the following questions about your environment:

- Do you have a local or remote database?
- How much disk space is allocated to the database?
- What tablespaces will be used for NetRanger data?

The answers to these questions will allow you to better install/mount Oracle.

Install/Mount the Oracle Product

You have two setup options for the Oracle Product: local or remote. Whether you use a local or remote setup, you will need to enter passwords for the Oracle users sys and system as part of the install process. Remember these passwords because you will use them when setting up the Oracle NetRanger user.

Local

Use the Oracle product CDROM and install with `orainst`. You will need Oracle Server (RDBMS) and SQL*Plus. Make sure to note the directory location of the install because you will use this in setting `$ORACLE_HOME`. Also note the SID, because you will use this to set `$ORACLE_SID`.

Remote

This is the standard client/server model, in which the server physically runs Oracle and the client obtains access to it through the DBMS network facilities. If you choose a remote setup for Oracle, you will need to choose one of the following setups:

- Remote homogeneous
- Remote heterogeneous

Remote homogeneous

If your client is the same operating system as the server, you can share the Oracle directory from the server, and mount it to the client (using normal NFS procedures).

Remote heterogeneous

If your client has a different operating system than the server, use the Oracle Installer with the product CDROM to install SQL*Plus and the TCP/IP Protocol adapter. Note the directory location of the install, because you will use this when setting `$ORACLE_HOME`. When the product binaries are available, you will need to configure tnsnames.ora, the Oracle network access file.

The tnsnames.ora file maps a label (such as remoteDB) to the access protocols to connect to it. A sample tnsnames.ora follows:

```
remoteDB =
   (DESCRIPTION=
      (LOCAL=NO)
      (ADDRESS=
         (PROTOCOL=TCP)
         (HOST=DBserver)
         (PORT=1521))
      (CONNECT_DATA=(SID=nr)))
```

Where `remoteDB =` is the label used in the connect string, i.e., username/password@remoteDB; where `HOST=Dbserver` is the network-assigned nodename of the server machine; and where `SID=nr` is the instance of Oracle running on the server, i.e., `$ORACLE_SID`.

Ensure that the tnsnames.ora file is correctly located and configured; otherwise, you will get a Oracle TNS error message. Oracle will look for it in the following filepaths:

```
/var/opt/oracle/tnsnames.ora

$ORACLE_HOME/network/admin/tnsnames.ora

$HOME/.tnsnames.ora
```

Setup Environment Variables

Oracle will not run without Environment variables. Environment variables are set differently depending on the setup you have chosen. Refer to the appropriate section for setting environment variables.

Remote or Local

Whether you are using a remote or local setup, Oracle requires the following three basic environment variables: `ORACLE_HOME`, `PATH`, and `LD_LIBRARY_PATH`.

`$ORACLE_HOME` is the base path for the Oracle product, such as /opt/app/oracle/product/7.1.6 or /usr/apps/oracle/app/product/7.3.2. `$ORACLE_HOME` usually ends in the version number of your Oracle product. The environment variable `PATH` must contain `$ORACLE_HOME/bin` to allow access to the Oracle binaries. The environment variable `LD_LIBRARY_PATH` must contain `$ORACLE_HOME/lib` to allow access to the runtime libraries.

If you have already set up a UNIX netrangr user, you can modify the .profile in /usr/nr and then re-source it to activate the updated variables (`ORACLE_HOME` is already set in the .profile, but you will probably need to change the directory path—use the supplied Oracle setup in .profile as a template).

However, if you have not set up a UNIX netrangr user, you can modify the .profile of the current user, or set the variables manually.

NOTE

Before nr.sapd can run, you must update the UNIX netrangr .profile to reflect your Oracle setup.

Local Only

If using a local setup, Oracle requires another environment variable, `ORACLE_SID`, which specifies what data area to use with a local database. When this is the case, update `ORACLE_SID` in the .profile as above.

tty problems (HP-UX)

On certain platforms, such as HP-UX, the default tty settings use the character @ for some built-in commands. You will have a problem connecting to a remote database because the tty settings will delete a portion of your connect string username/password@remoteDB.

To examine the tty setting, execute the following command:

```
% stty -a
```

This will show the mappings for commands. If one of the commands, such as kill, is mapped to @, you will need to change it by executing the following command:

```
% stty kill ^Z
```

Verify Generic Connectivity

Verify that Oracle access is enabled.

Before attempting an Oracle connection, ensure that `$ORACLE_HOME`, `$PATH`, and `$LD_LIBRARY_PATH` are set correctly (as described above).

Local

Set `$ORACLE_SID`. The Oracle daemons must be running. Use sqlplus to verify basic connectivity to Oracle with the following command:

```
% sqlplus sys/password
```

If successful, you will enter into interactive SQL. You will see a prompt similar to the following:

```
Connected to:
Oracle7 Server Release 7.3.2.1.0 - Production Release
PL/SQL Release 2.3.2.0.0 - Production
SQL>
```

NOTE

You can omit the password string in the command line call and sqlplus will prompt you for the password. This is better for security because the password does not show up in ps output.

Remote

Use tnsping to verify that the network setup is correct (this is similar to the standard ping utility) with the following command:

```
% tnsping remoteDB
```

Use sqlplus to verify basic connectivity to a remote Oracle server (similar to the local sqlplus verification, but the connect string will contain `@DBserver`) with the following command:

```
% sqlplus sys/password@remoteDB
```


Refer to the Appendix A for information about the common error messages you might encounter.

Create User

Follow these steps before you create a user:

1. Log into Oracle as sys or system (using sqlplus or sqldba).
2. Determine what Oracle data areas (tablespace) will be assigned for NetRanger usage. This tablespace needs to be as large as possible. The option of assigning a default tablespace to a user eliminates the problem of having to direct subsequent commands to use a certain tablespace. For high volume NSX sensors, you may need to dedicate a TEMPORARY TABLESPACE and a larger ROLLBACK area to the NetRanger user. This is also done in the create user command.

You are now ready to create a user. Follow these steps to create a user:

1. Create the user with the following command:

```
SQL>  create user NetRanger identified by PASSWORD default tablespace 'USERS'
```

Where PASSWORD is your secret password and where USERS is your designated tablespace.

2. Activate the account with the following command:

   `SQL> grant dba to NetRanger` (or other lesser privilege based on your site's security policy; at a minimum, the NetRanger user will need privilege to CONNECT, SELECT, INSERT, and DELETE)

IMPORTANT NOTE

Make sure to remember the user and password. You will need to update the Tokens DBUser and DBPass to reflect the new user and password. This can be done either by editing the file /usr/nr/etc/sapd.conf or by using nrset with the tokens DBUser and DBPass. *sapd* should not be activated until these are set (unless you are customizing the scripts).

Verify NetRanger–Oracle Connectivity

Verify that the UNIX netrangr user can access the Oracle NetRanger account at this time.

As a UNIX netrangr user on the machine where nr.sapd will be run, use sqlplus (as described above) to connect to Oracle as the Oracle NetRanger user (created previously).

If you get a SQL> prompt, proceed to the next step. If you get an error, go back to the previous step(s).

Create Tables

In the previous step, you verified that the UNIX netrangr user has access to the Oracle NetRanger account. The last step is setting up the database tables, which will be populated by the log file data.

These database tables must be created before *sapd* can start loading the data. As UNIX user netrangr, use sqlplus to connect to Oracle and create the tables.

Once connected, execute the following commands at the SQL> prompt:

```
SQL> @/usr/nr/bin/sap/skel/create_log_alarm
SQL> @/usr/nr/bin/sap/skel/create_log_err
SQL> @/usr/nr/bin/sap/skel/create_log_cmd
SQL> @/usr/nr/bin/sap/skel/create_sigs
```

NOTE

See *Using NetRanger with RDBMS* for information on type 5 sapx loads in `create_sigs`.

Do not worry if you see an error message that says the table does not exist. This is normal on the first install.

Be careful when using these create scripts, because they will drop the existing table, thus deleting all the data.

NOTE

Refer to the *sapd* documentation if you wish to customize the database schema.

Setup for Remedy ARS (Optional)

Use schema provided in /usr/nr/bin/sap/skel/nr.alarm.schema.def. See the *Using sapx with Remedy* section for more information about Remedy ARS.

Non-Oracle Install and Setup

Please refer to your DBMS software documentation for installation and setup. Use the *Oracle Install and Setup* section as a template.

DBMS SETUP CHECKLIST

The following checklist aids in setting up a DBMS. It also provides a high-level overview for Appendix B, in which can be found full details of what follows.

Install and Configure Oracle

_____ Install Oracle Product: CD-ROM or NFS mount

_____ Configure .profile: `ORACLE_HOME, PATH, LD_LIBRARY_PATH`

_____ Set up SQLNET: tnsnames.ora

_____ Verify SQLNET: sqlplus or tnsping

NetRanger–Oracle Setup

_____ Create user

_____ Verify: UNIX netrangr user login to Oracle as netranger

_____ Create tables

nr.sapd Setup

_____ Configure /usr/nr/etc/sapd.conf: set DBUser and DBPass

_____ Configure /usr/nr/etc/daemons: add nr.sapd to list

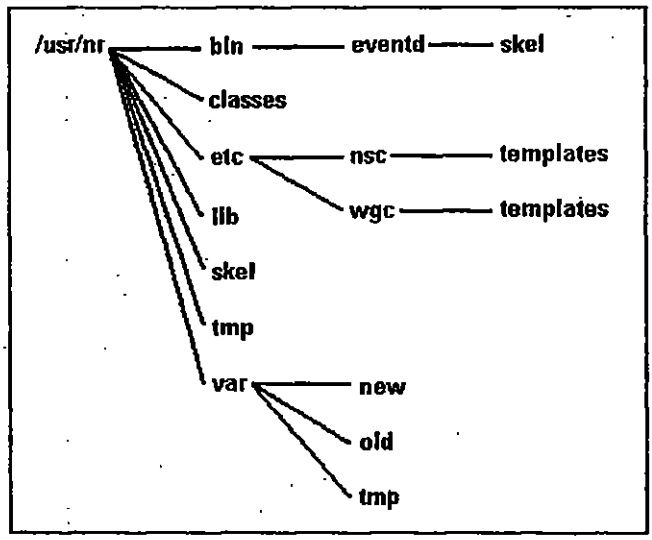
_____ Execute: `nrstart`

_____ Verify that *sapd* is running: `tail -f /usr/nr/var/messages.sapd`

Appendix C The NSX File Structure

One of the quickest ways to gain an understanding of NetRanger is to review the NSX file structure and underlying elements. A thorough knowledge of the NSX file structure will help you configure and install the NSX as well as help you better utilize NetRanger's features. It should be noted that this file structure serves as the foundation for both the Director and NSX systems.

The default directory location for the NSX subsystem is /usr/nr, which contains the following directory structure:



This appendix provides a brief explanation of each of these directories.

/bin

This directory contains all of the executables required to run and administer the application services that support an NSX or Director system. These can be loosely grouped into three basic categories:

- Daemon Applications
- Configuration Commands
- System Commands

Daemon Applications

As the *Overview* states, NetRanger is a collection of discrete subsystems that have been implemented via daemons, each with its own well-defined set of services.

NetRanger is also a highly configurable distributed application. The daemon services running on a NSX or Director host depends on the configuration of the application as a whole. By default, a NSX host must have *sensord, managed, loggerd,* and *postofficed* daemons running. A Director system should have *postofficed, smid, loggerd,* and *configd* running. In a multi-tiered configuration, one Director system could be configured to only stage data to an RDBMS via *loggerd and sapd,* while another Director system could be dedicated to monitoring and response via the *smid* and *configd* daemons.

The full complement of NSX daemons is as follows:

- nr.configd
- nr.eventd
- nr.loggerd
- nr.managed
- nr.postofficed
- nr.sapd
- nr.smid
- nr.sensord

Each of these daemon services is briefly described below.

configd

This daemon reads and writes data to the NSX configuration files (which are currently restricted to ~/etc). All communication with this daemon is controlled by the following set of *SNMP-like* executables:

- nrget & nrgetbulk
- nrset & nrunset
- nrexec

NOTE

- This daemon provides *an interface* to configuration information. Actual reconfiguration is performed by *managed.*
- Both command line and graphical user interfaces are provided.

eventd

This daemon is designed to run on Director systems. It allows the Director system to generate user-defined actions for events received by *smid*. A common action is to generate pager notifications via e-mail for alarms of severity "4" and above.

loggerd

This daemon writes out sensor and error data to flat files generated by one or more of the other daemon services. This data is passed to *loggerd* via *postofficed*. *loggerd* creates two basic types of flat files: a single NSX Event file, and one or more session logs. As noted in the *Overview*, data is written to flat files for reasons of performance and fault tolerance. This data can be staged onto a RDBMS via *sapd*.

managed

Whereas *sensord* is responsible for interpreting packet filter events, *managed* is responsible for managing and monitoring these devices. For example, when *sensord* identifies that a certain type of attack should be shunned, it sends a shun command to *managed* via the post office facility. *managed* then reconfigures the packet filter appropriately. Similar commands can be sent to this daemon from the Director. *managed* can also be polled for operational statistics such as:

- the number of network devices being managed
- device type
- device statistics (packets/sec, bytes/sec, etc.)

postofficed

This daemon serves as the communication vehicle for the entire NetRanger product. All communication between daemons is routed through this service. In most NetRanger configurations the NSX and Director systems reside on separate hosts. Each system relies on a *postofficed* to maintain communication with the other. Note that a *postofficed* is required even when all of the systems are located on a single host.

This daemon service include the following features:

- A proprietary connection-based protocol that resends a message whenever any of its packets are lost.
- Point-to-point routes that may include intermediate post office nodes.
- Communication integrity that is maintained via alternate routes (specified in ~/etc/routes and ~/etc/destinations). When one route fails, communication is switched to an alternate route. Communication is reestablished to the preferred route as soon as it comes back online.

Routing is based on a three part key that includes the following tuples:

- Organization
- Host
- Application

The Organization IDs are defined in ~/etc/organizations; Host IDs are defined in ~/etc/hosts; Application IDs are defined in ~/etc/services.

sapd

This daemon is a light weight scheduler that pulls data from event log files and exports it into either an Oracle or Remedy database.

smid

This daemon routes messages to and from the Director and other daemon services, such as *sensord.* This service relies on routing information stored in the Director's ~/etc/hosts file. In addition to being able to communicate with other daemons, *smid* can redirect messages to other daemon services, such as eventd and loggerd based on *DupDestination* entries in etc/smid.conf.

sensord

This daemon interprets and responds to all of the events generated by one or more packet filter devices. Data is read in from Network Device sockets and sent to *postofficed* for routing to other daemon services. The types of events received by sensord is a function of the filter policy on each packet filter device. The types of messages *sensord* sends to other daemons is based on information stored in ~/etc/sensord.conf. Where these messages are routed is defined in ~/etc/destinations.

Although *sensord* supports 256 levels of alarm events (0-255), only levels 0-6 are currently being used. Level "0" is an internal alarm that *sensord* uses to track such services as RIP, ICMP, and so on; it is never routed to *postofficed.* Levels 1-6 are sent to *postofficed,* and they identify increasing patterns of misuse, where "5" is the highest type of alarm. Note that every alarm message includes an Alarm-Id that identifies a specific attack signature. These alarm signatures are enumerated in ~/etc/sensord.conf as SigOfGeneral entries.

In addition to generating alarm messages for a Director application, *sensord* can automatically instruct *managed* to shun an attack for a specified period of time.

Configuration Commands

The NSX Sensor system comes with the following built-in commands that allow remote devices to configure and gather information about the NSX. As noted in *Operating the Director*, these commands serve as the foundation for the Director's **nrConfigure** interface.

- nrget
- nrgetbulk
- nrset
- nrunset
- nrexec

nrget

This file retrieves single pieces of information from the NSX.

```
nrget socket appid hostid orgid priority token [ identifier... ]
```

nrgetbulk

This file retrieves multiple pieces of information from the NSX.

```
nrget socket appid hostid orgid priority token [ identifier... ]
```

nrset

This file sets attributes about the NSX.

```
nrset socket appid hostid orgid priority token [ identifier...] value [value ...]
```

nrunset

This file unsets attributes about the NSX.

```
nrset socket appid hostid orgid priority token [ identifier... ] value [value ... ]
```

nrexec

This file executes commands on the NSX.

```
nrset socket appid hostid orgid priority token [ identifier... ] value [value ... ]
```

System Commands

The following scripts are used to perform maintenance on the NSX:

- install
- nrstart
- nrstop
- nrstatus

install

This script is used to install the startup/shutdown files for NetRanger into the UNIX /etc directory. The command takes the following form:

```
install { add | remove }
```

nrstart

This script starts the NSX. It supports the following options:

- -d Don't daemonize
- -h Show help
- -v Show version number

The command takes the following form:

```
nrstart [-d] [-h] [-v]
```

nrstop

Executing this script stops the NSX.

nrstatus

Executing this script returns the currently running daemons.

/Classes

This directory serves as the *rooted class path* for the Director's Java applications, such as **nrConfigure.**

/etc

This directory contains two types of configuration files: **daemon-specific** and **NSX system** files. All of NetRanger's configuration files are currently ASCII flat files.

Daemon Configuration Files

There is a one-to-one correspondence between daemons and their configuration files. Whereas the naming convention for a daemon is **nr.<daemon>**, the convention for its configuration file is **<daemon>.conf**. Each of these files contains token-based configuration information of the form: **<token> <values>**. For example, the error file for nr.managed is identified in managed.conf as follows:

```
FilenameOfError    ../var/errors.managed
```

Although these entries initially can be set via a text editor, all subsequent access should be managed via the nr utilities found in **~/etc**.

- configd.conf
- eventd.conf
- loggerd.conf
- managed.conf
- postofficed.conf
- sapd.conf
- sensord.conf
- smid.conf

NSX System Configuration Files

The **~/etc** system configuration files are modeled after standard UNIX /etc files and currently consist of the following files.

- auths
- daemons
- destinations
- hosts
- organizations
- routes

- services
- signatures

~/skel

This directory contains basic UNIX configuration files (such as .profile) that are required to configure remote login accounts.

~/tmp

This is the directory where sockets are created by the different daemons. The names of these sockets are dictated by the parent daemon. *postofficed* creates sockets with names such as mailbox.sensord and mailbox.configd. *configd* creates sockets with names such as socket.command.

~/var

This is the default location for all of the log files generated by *loggerd* and error files for all of the applications listed in the ~/etc/daemons file.

Log Files

Two different types of log files are created by *loggerd:* **Event** and **IP Session** logs.

Event File

Although the name of this file can be changed via nrset, the default name of this file is currently hard-coded in *loggerd*. The name of event logs are based on the date and time a new log began (e.g., log.199607291332). Although even message types are defined, only three are currently implemented.

Message Type	Description
0	Default
1	Command
2	Error (Defined)
3	Command Log (Defined)
4	Event (Defined)
5	IP Log
6	Redirect

The schema for defined message types are as follows:

Error records from *managed* (ERROR)

```
1, MsgType (SInt) nrPROTOCOL_ERROR,
2, ULong RecordID,
3,4,5 timestamp ( ULong time, String year/month/day,hour:min:sec )
6, ULong addr.ApplID,
7, ULong addr.HostID,
8, ULong addr.OrgID,
9, String ErrorData
```

Command Log records from *configd* (COMMANDLOG)

```
1, MsgType (SInt) nrPROTOCOL_COMMANDLOG,
2, ULong RecordID,
3,4,5 timestamp ( ULong time, String year/month/day,hour:min:sec )
6, ULong addr.ApplID,
7, ULong addr.HostID,
8, ULong addr.OrgID,
9, ULong src.ApplID,
10, ULong src.HostID,
11, ULong src.OrgID,
12, String LogData
```

Event/Alarm records from *sensord* (PROTOCOL_EVENT)

```
1, MsgType (SInt) nrPROTOCOL_EVENT,
2, ULong RecordID,
3,4,5 timestamp ( ULong time, String year/month/day,hour:min:sec )
6, ULong addr.ApplID,
7, ULong addr.HostID,
8, ULong addr.OrgID,
9, String from,
10, String to,
11, SInt event->level,
12, ULong event->SigID,
13, ULong event->SubSigID,
14, String protocol,
15, String sourceIpAddress,
16, String destIpAddress,
17, SInt event->SrcPort,
18, SInt event->DstPort,
19, String routerIpAddress,
20, String EventData
```

Session Log Files

In addition to detecting attack signatures, *sensord* is able to monitor the UDP traffic associated with a specific type of attack. For example, *sensord* can be configured to monitor all of the packets associated with an IP spoof. *loggerd* creates a separate log file in **~/var** for each of these monitoring sessions. The name of each session log file is based on the IP address of the attacking host (e.g., log.209.15.163.54).

Appendix D NetRanger Hardware Components

NSX Sensor Hardware Components

- Rack-Mounted PC w/Solaris x86 2.5
- 166 MHz Pentium® processor (10Mbps)
- 110 Mhz UltraSPARC (100Mbps)
  - ◊ wide SCSI
- 32 MB RAM
- Three NSX Versions:
  - ◊ NSX 1000
  - ◊ NSX 2000
  - ◊ NSX 5000
- Four configuration options:
  - ◊ access through serial port
  - ◊ network access
  - ◊ modem dial-up
  - ◊ video/keyboard

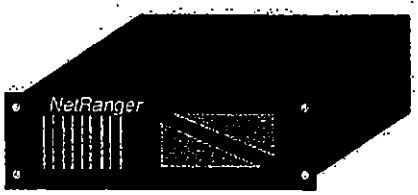


Figure D-1: The NSX Hardware Component of the NetRanger System

NSX Versions

- *NSX 1000*
  - ◊ Pentium 166 NSX
  - ◊ NSC BorderGuard 1000
    - ⇒ 2 10BaseT / AUI Ethernet
    - ⇒ 3 WAN Daughtercards
  - ◊ Up to 512 Kbps

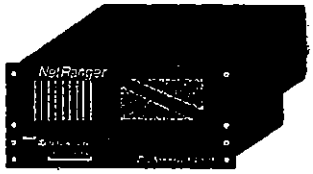


Figure D-2: The NSX 1000

- *NSX 2000*
  - ◊ Pentium 166 NSX
  - ◊ NSC BorderGuard 2000
    - ⇒ 4 Ethernet
    - ⇒ 2 Ethernet / 2 WAN
  - ◊ 512 Kbps to 10 Mbps Ethernet

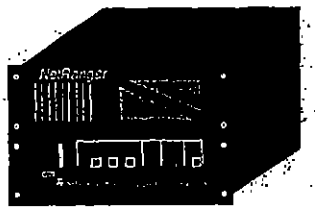


Figure D-3: The NSX 2000

- *NSX 5000*
  - ◊ UltraSPARC NSX
  - ◊ NSC ERS/Passport
    - ⇒ FDDI
    - ⇒ Ethernet
    - ⇒ WAN
    - ⇒ (ATM)
  - ◊ 10 to 100 Mbps Ethernet
  - ◊ Voice as well as Data

Figure D-4: The NSX 5000

NSC BorderGuard Hardware Components

- The BorderGuard is a bridge/router that links IEEE 802.3/Ethernet networks, and other networks via WAN links (e.g., ISDN, leased lines).
- Includes Packet Control Facility (PCF) for firewall routing
- Includes Bridge Control Facility (BCF) for filtered bridging
- Routing support
  - ◊ TCP/IP
  - ◊ DECnet Phase IV
  - ◊ Novell IPX
  - ◊ Banyan VINES
  - ◊ AppleTalk II
  - ◊ XNS

Figure D-5: The NSC BorderGuard

Appendix E Uninstalling the Director

Uninstallation Process for HP-UX Systems

This section describes how to remove the Director software from an HP-UX system. Please note that this does not describe how to remove OpenView itself—HP OpenView will still be installed after executing the steps in this section. If you want to remove OpenView as well as the Director, uninstall the Director first, then consult the HP OpenView documentation to remove OpenView.

NOTE

Note: If you want to uninstall the Director but continue to use HP OpenView, you might consider deleting your NetRanger/Director data from the OpenView databases *before* uninstalling the Director. **Once you uninstall the Director, there will be no way to remove the data from the databases, other than by completely removing the databases.** To delete data from the databases while the Director is still installed, use the **Delete Object** menu item from the user interface.

To uninstall the part of the NetRanger/Director that integrates with OpenView (again, this does not uninstall OpenView), follow these steps:

1. **Stop the NetRanger daemons using this command:**

   ```
   /usr/nr/bin/nrstop
   ```

2. **Login as "root" if you have not already done so.**
3. **Instruct all users who are logged on as user "netrangr" to log off now.**
4. **From the command line, type:**

   ```
   /usr/sbin/swremove
   ```

Uninstalling the Network Management Interface software

1. **From the list of applications, select** WGCdrctr.
2. **From the** Actions **menu, select** Mark for Remove.
3. **From the** Actions **menu, select** Remove.
4. **A pop-up window will appear notifying you that analysis is being done.**
5. **When the analysis completes, press** OK.
6. **Press** Yes **to start the removal.**
7. **Press** Done **when the removal is complete.**

Uninstalling the Remote Configuration software

1. From the list of applications, select WGCcfg.
2. From the Actions menu, select Mark for Remove.
3. From the Actions menu, select Remove.
4. A pop-up window will appear notifying you that analysis is being done.
5. When the analysis completes, press OK.
6. Press Yes to start the removal.
7. Press Done when the removal is complete.

Uninstalling the RDBMS software

1. From the list of applications, select WGCsapd.
2. From the Actions menu, select Mark for Remove.
3. From the Actions menu, select Remove.
4. A popup window will appear saying that analysis is being done.
5. When the analysis completes, press OK.
6. Press Yes to start the removal.
7. Press Done when the removal is complete.

Uninstalling the NSX Interface software

1. From the list of applications, select WGCnsx.
2. From the Actions menu, select Mark for Remove.
3. From the Actions menu, select Remove.
4. A popup window will appear saying that analysis is being done.
5. When the analysis completes, press OK.
6. Press Yes to start the removal.
7. Press Done when the removal is complete.
8. Exit the swremove program.

To remove the **netrangr** userid, follow these steps:

1. On HP systems, bring up the SAM utility by typing the following:
   ```
   sam &
   ```
2. Select Accounts for Users and Groups.
3. Double-click the Users icon.
4. Select netrangr from the list.
5. From the Actions menu, select Remove.

6. **Determine whether or not you want all of the files owned by** netrangr **to be deleted, and make the appropriate selection in the dialog box.**
7. **Press** OK.
8. **Press** Yes **to remove the user.**
9. **Press** OK **when the process is complete:**

To remove the **netrangr** group (HP-UX only), follow these steps:

1. **Double-click on the** Groups **icon.**
2. **Select** netrangr **from the list.**
3. **From the** Actions **menu, select** Remove.
4. **Determine how files that belong to this group should be treated, and make the appropriate selection in the dialog box.**
5. **Press** OK.
6. **Press** Yes **to remove the group.**
7. **Press** OK.

Uninstallation Process for Sun Solaris Systems

This section describes how to remove the Director software from a Sun Solaris system. Please note that this does not describe how to remove Solaris itself—Solaris will still be installed after executing the steps in this section. If you want to remove HP OpenView as well as the Director, uninstall the Director first, then consult the Sun Solaris documentation to remove HPOpenview.

NOTE

Note: If you want to uninstall the Director but continue to use HP OpenView, you might consider deleting your NetRanger/Director data from the Solaris databases *before* uninstalling the Director. **Once you uninstall the Director, there will be no way to remove the data from the databases, other than by completely removing the databases.** To delete data from the databases while the Director is still installed, use the **Delete Object** menu item from the user interface.

To uninstall the part of the NetRanger/Director that integrates with HP OpenView (again, this does not uninstall HP OpenView), follow these steps:

1. **Login as root, or use the "su" command to become the root user.**
2. **Type the following command to remove the Director user interface application:**

   ```
   pkgrm WGCdrctr
   ```

3. **Type the following command to remove the RDBMS software:**

   ```
   pkgrm WGCsapd
   ```

4. **To remove the Remote Configuration software, type this command.**

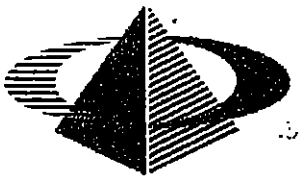
   ```
   pkgrm WGCcfg
   ```

5. **Type the following command to remove the NSX Interface software:**

   ```
   pkgrm WGCnsX
   ```

6. **Remove the netrangr user id (optional) with the following commands:**

   ```
   userdel -r netrangr
   groupdel netrangr
   ```

WheelGroup Corporation

WheelGroup Corporation welcomes your ideas on how to improve our documentation. Please take a moment to answer the questions below by checking either the "Yes" or "No" box. Explain any "No" responses in the sections provided. Include other comments in the "Comments" section. When you have finished, please detach and mail this page, or fax it to (210) 494-6303. You may also e-mail us about documentation issues at documentation@wheelgroup.com.

Does this manual contain all the information you expected?	◊ Yes	◊ No
Are all procedures documented in the manual correct?	◊ Yes	◊ No
Is this manual easy to read and understand?	◊ Yes	◊ No
Are examples and diagrams helpful?	◊ Yes	◊ No
Are there enough examples and diagrams?	◊ Yes	◊ No
Other comments:		

CERTIFICATE OF SERVICE

I hereby certify that on the 30th day of June, 2006, I electronically filed the foregoing document, **DECLARATION OF PAUL S. GREWAL IN SUPPORT OF SYMANTEC CORPORATION'S OPPOSITION TO SRI INTERNATIONAL, INC.'S MOTION TO EXCLUDE FROM EVIDENCE THE EXPERT OPINION OF DANIEL TEAL, VOLUME 2 OF 4,** with the Clerk of the Court using CM/ECF which will send notification of such filing to the following:

John F. Horvath, Esq.
Fish & Richardson, P.C.
919 North Market Street, Suite 1100
Wilmington, DE 19801

Richard L. Horwitz, Esq.
David E. Moore, Esq.
Potter Anderson & Corroon LLP
Hercules Plaza
1313 North Market Street, 6th Floor
Wilmington, DE 19801

Additionally, I hereby certify that on the 30th day of June, 2006, the foregoing document was served via email and by Federal Express on the following non-registered participants:

Howard G. Pollack, Esq.
Michael J. Curley, Esq.
Fish & Richardson
500 Arguello Street, Suite 500
Redwood City, CA 94063
650.839.5070

Holmes Hawkins, III, Esq.
King & Spalding
191 Peachtree Street
Atlanta, GA 30303
404.572.4600

Theresa Moehlman, Esq.
King & Spalding LLP
1185 Avenue of the Americas
New York, NY 10036-4003
212.556.2100

/s/ Richard K. Herrmann

Richard K. Herrmann (#405)
Mary B. Matterer (#2696)
Morris, James, Hitchens & Williams LLP
222 Delaware Avenue, 10th Floor
Wilmington, DE 19801
(302) 888-6800
rherrmann@morrisjames.com

Counsel for Symantec Corporation